ISSUE 12 - MAY 2013

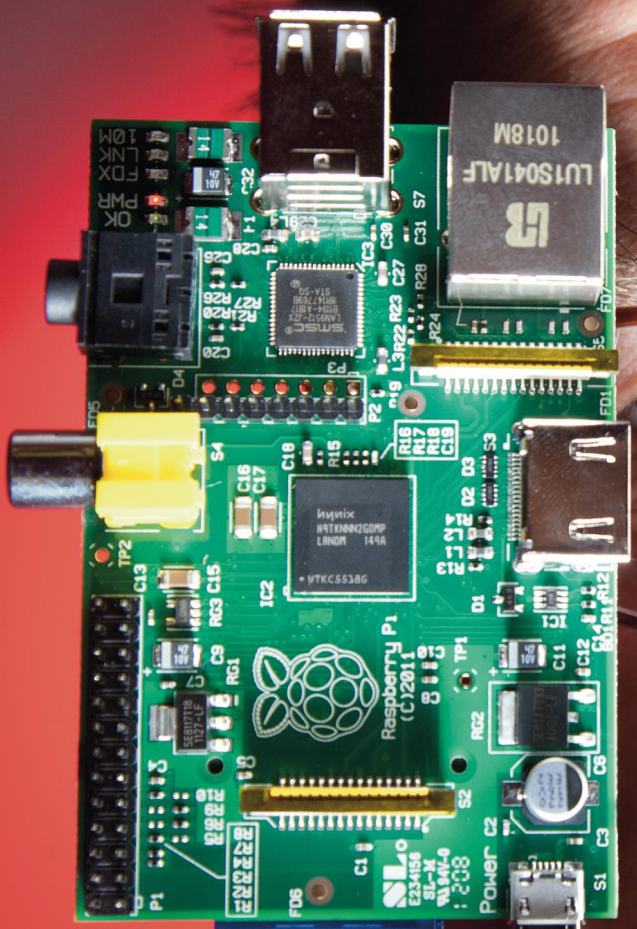
Get printed copies at themagpi.com

The MagPi

A Magazine for Raspberry Pi Users

This Issue...

- A Year Of The MagPi with Liz Upton
- SchismTracker: DJ Quicksilver
- Encryption With Scratch
- Pete Lomas Interview



BIG BIRTHDAY COMPETITION
Win more than **£1,000** of goodies!

ISSN 2051-9990

Raspberry Pi is a trademark of The Raspberry Pi Foundation.
This magazine was created using a Raspberry Pi computer.

CC BY NC SA

The MagPi

http://www.themagpi.com

The MagPi 12

Welcome to issue 12 of The MagPi magazine, our 1st birthday celebratory issue.

What a year it has been! To mark this incredible achievement Liz Upton from the Raspberry Pi Foundation has kindly agreed to be a guest writer for the magazine and has produced an article documenting our first year of publication. Thank you Liz.

To celebrate a year of The MagPi, in this issue we are proud to be able to hold our first massive birthday give-away. In this competition we are giving readers the opportunity to win over £1,000 of Raspberry Pi goodies! Thank you to all the companies who have supported us with this... you are amazing.

We also take this opportunity to include articles covering some of the less published start up questions in the 'Beginner's Guide' article and we review some of the operating systems which were produced over the last year for the Raspberry Pi. The MagPi's very own Ian McAlpine provides us with a fantastic insight into the mind of Pete Lomas plus more from our great authors on coding.

Thank you to those who supported our Kickstarter campaign. We have a special update on page 17. Expect your printed copies very soon!

I hope you continue to enjoy The MagPi into our next year of production. If you have an article you wish us to include, or have some free time you could spare to help our cause, please email us at editor@themagpi.com.

Chief Editor of The MagPi

The MagPi Team

Ash Stone - Chief Editor / Administration
W.H. Bell - Layout / Graphics / Administration
Matthew Judge - Website / Administration / Tester
Chris 'tzj' Stagg - Administration / Tester / Graphics
Tim 'meltwater' Cox - Administration / Layout Tester
Lix - Administration
Ian McAlpine - Layout / Tester / Graphics
Aaron Shaw - Layout / Graphics / Proof Reading
Bryan Butler - Page Design & Theme / Graphics / Tester
David Allan - Tester
Sam Marshall - Page Design
Colin Deady - Layout / Graphics
Mark Tranter - Layout
James Nelson - Layout / Graphics
Steve Drew - Layout
Matthew Watson - Layout
Adrian Harper - Layout / Proof Reading
Phil Tesseyman - Tester
Shelton Caruthers - Tester / Proof Reading
Sai Yamanooor - Tester / Proof Reading
Jeremy Schrock - Proof Reading
Chad Longstaff - Proof Reading
Phil Shillito - Proof Reading

Contents

4 **A YEAR OF THE MAGPI**
Liz Upton writes about the first year of The MagPi

6 **BEGINNERS GUIDE**
Where can I get help?

9 **THIS MONTH'S EVENTS GUIDE**
Malawi, Swansea, Tokyo and Norwich

10 **SWEETBOX II**
How to make a case for the Raspberry Pi in 314 steps

12 **RASPBERRY PI OPERATING SYSTEMS**
A breakdown of the various operating systems available for the Pi

16 **STAR LETTER**
A note of thanks from the ZX generation

18 **PRINTING WITH CUPS 2**
Printing from a Python program

20 **AN INTERVIEW WITH PETE LOMAS**
Sharing his thoughts on Raspberry Pi design challenges and the future

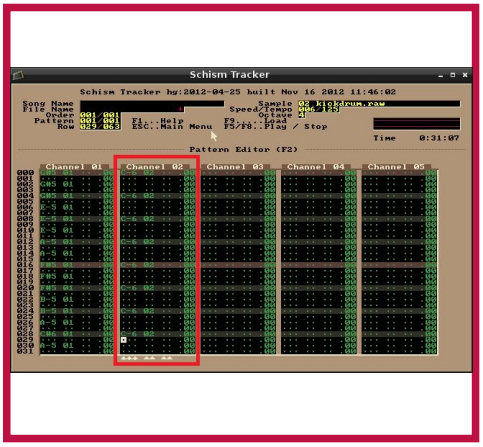
26 **MAKING MUSIC WITH SCHISM**
Programming melodies

30 **COMPETITION**
A birthday issue bonanza of goodies to be won, worth over £1,000!

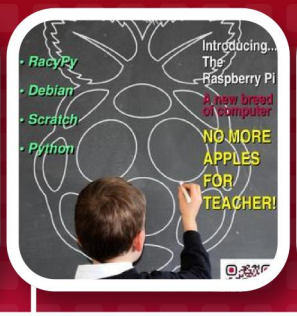
32 **BASH GAFFER TAPE**
Embedding text and programs

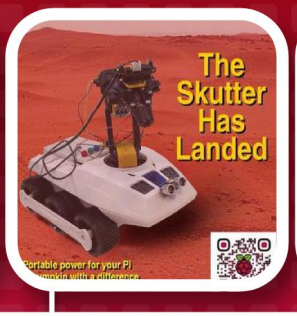
34 **SCRATCH PATCH**
Use encryption to code and decode messages

39 **FEEDBACK**
Have your say about The MagPi



Cover photo plus photo on page 20 by Dean Smith, Camera Crew.





A Year of The MagPi

Feb 2012 | May 2012 | Oct 2012 | Dec 2012

Looking back at the first year of The MagPi with Liz Upton

Liz Upton

The Raspberry Pi Foundation

Last March, an email came through to my Raspberry Pi Foundation address from someone calling himself Jaseman. He and a group of volunteers had been working on a monthly Raspberry Pi-themed magazine, and they were intending to give away electronic copies for free.

There had been chatter on the Raspberry Pi forums about what a nice idea a monthly magazine, produced by Pi users, would be; a sort of return to the old days of reviews, listings and hardware projects that some of us remembered from magazines like Micro User and Amiga Format. I didn't for a moment think that anybody would take that suggestion and turn it into a real magazine; this was just after the Pi's launch announcement and we hadn't really got a feel for just how dynamic the Pi community can be when it comes up with a really good idea.

> *"I can't imagine doing all this stuff with a team of volunteers"*

Magazines, you see, are hard. Before I started working for Raspberry Pi in 2011 I'd spent ten years working in publishing; first as an assistant editor for a group of classical music magazines, before moving into book publishing as a chief copy editor. I moved to books for a simple reason: magazines give you sleepless nights. The pay is awful; you'll find yourself subsisting on the canapés they serve up at events so you don't have to buy food. You're working on a month-long treadmill that never stops moving. You're corralling a million different contributors, none of whom take your deadline seriously; you're ensuring that everything gets proof-read by three different sets of eyes before going to press and trying to remember where the accents go in Antonín Dvořák; you're hoping that your advertising department manages to raise enough money this month to keep you alive for another cycle; you're chatting up the production department on the off chance that they'll be able to create a bit more space for the really great feature that turned out too long for this issue; you're thrashing about in an attempt to come up with a really clever headline for that article about the old oboe da cacca-playing gent with the already-too-punny name; and then, when the thing comes out, you'd better brace yourself for the reader feedback. Letters will arrive written in purple ink informing you that the theorbo you pictured on page 13 was, in fact, a modern sitar, and calling for your sacking. (It was a theorbo. I wasn't sacked.)

Seriously: if you think that electronics enthusiasts are geeks, you ought to meet the early music crowd.

I can't imagine doing all this stuff with a team of volunteers, absent the threat of withholding someone's fee if they don't

pitch up on time with their article. I really can't imagine doing it without a team of experienced layout, editorial, production, photography, advertising, financial and legal people propping the whole enterprise up. But we kept our fingers crossed and hoped The MagPi team would meet with the same sort of luck we'd found at the Foundation and we offered them our help if they felt they needed it. (They didn't.) The core team seemed like a smart bunch of people. We prepared ourselves to be surprised.

And we were. The MagPi's quality, convincing from the start, has really been cemented over the last year. Graphically it's getting better and better with every issue, and the contents are so good that the Raspberry Pi Foundation has been strongly recommending the magazine to schools, parents and after-school clubs to use as part of their teaching materials. The team working on the magazine has somehow found time not only to get a magazine out every month, but also to run a startlingly successful Kickstarter and, in the case of people like Tim "Meltwater" Cox, to set up some small businesses in the Pi ecosystem as well. I've referred people emailing me with questions to the magazine time and time again. It's a superb resource, and it's been driven by nothing but the extraordinary enthusiasm of a group of spare-time volunteers.

I met Ash Stone, who edits The MagPi, at an Agile conference in Cornwall last summer. We spent the evening sitting on the beach in Falmouth with him, drinking beer and eating hogroast, while talking about the magazine. He has a very demanding day job, but brings an enthusiasm and a work ethic to his job on The MagPi that mirrors exactly what we saw at the Raspberry Pi Foundation in the days when we were a group of volunteers too. We have found that working for the love of something brings rewards that the daily grind for a paycheck can't touch; Ash and the team seem to be experiencing something very similar. There's such a depth and breadth of electronics and programming knowledge inherent in the group that's come together to make this magazine. Their impulse to share that knowledge and make it accessible to everybody, not just those who have the luxury of being able to afford a monthly subscription, is really admirable and, we hope, reflects the philosophy behind the Raspberry Pi Foundation itself. We believe that access to tools should be something that's available to everybody. We believe that nobody should be priced out of being able to learn, or being able to build. Of course, computing and making is not for everybody; but everybody should be offered the opportunity to find out whether it's something they respond to or not.

We think there is something compelling, even in a period of ubiquitous e-readers and tablets, about the printed word. We're proud and excited to see The MagPi's print editions take off; even if you're not buying a print edition, I hope you're printing at least some of the PDF version magazine at home (or at work or school) every month. When you read a piece of code and then use your brain to process it and send a signal to your fingers to type it into your Pi, you're cementing what you're learning in a physical way that just won't happen if you use copy and paste. You will also, hopefully, make mistakes in your typing, which will mean you'll have to go over the code again to pull out bugs, which will make you think about how things work. If you're like me, you'll fold the corners of pages down and mark interesting bits up with a pen, or stick a post-it note on articles you want to come back to. Scribbling your own notes on hardware instructions helps you to remember what they're for and how things fit together. We're looking forward to seeing how the team deals with the challenges of producing a physical magazine alongside the PDF MagPi; we're also looking forward to seeing sets of the print MagPi in school classrooms and libraries.

"We think there is something compelling about the printed word"

This second year of The MagPi has potential to be even greater than the first. Everybody here at the Raspberry Pi Foundation would like to congratulate The MagPi team on a wonderful year; but most of all we would like to thank you. You've helped spread the word about our little computer; you've helped people to learn how to use it. You've helped kids to find their first enthusiasm for computing and engineering and you've helped grownups get back into a hobby they dropped when machines like the Amiga and the BBC micro went away. You're a vital part of an ecosystem that the Pi needs to stay important and relevant in education and we can hold you up as an example of the passion and skill of the community around the Raspberry Pi.

In many ways we wouldn't be where we are today without The MagPi. So thanks to all of you from all of us: and here's to year two!

Where can I get help?

Adrian Harper
MagPi Writer

DIFFICULTY : BEGINNER

There are numerous guides to getting started with a Raspberry Pi that cover the basics of setting up your new computer with the recommended Raspbian Wheezy operating system, running through raspi-config etc. However, most setup guides stop once the system boots into the windows style interface of LXDE.

I regularly encourage people to buy and use Raspberry Pis, especially if they have kids, but time and again I hear people asking *"Ok I'm at the desktop, what now?*", and *"How do I do such and such?*".

The official forums at raspberrypi.org are a great place to start looking for solutions to problems or questions that you might have, though they can be a bit daunting for a beginner. The search functions on the forum are excellent, so please take a few minutes to search the existing posts for answers to your questions before starting a new thread.

The web offers some exceptional free user guides that go into great detail to cover the installation of Raspbian Wheezy on an SD card from Windows, OSX and Linux. Engadget's Getting Started Guide is especially good - **http://engt.co/PZbUpT**. These guides offer help with troubleshooting basic issues as well as providing ideas for that "*What now?*" moment when everything is running and you're staring at a large raspberry in the middle of the screen. The Raspberry Pi Guide on the Embedded Linux wiki is also recommended - **http://bit.ly/pJfgMr**.

Even with the user guides and forums, it is worth mentioning the solutions to a few of the most common questions asked by new Raspberry Pi owners:

I would like to connect x device to my Raspberry Pi. How can I be sure it is going to work?

Unless you like the challenge of writing drivers and figuring out why a piece of hardware connected to your computer does not function as it should, before buying any peripherals for your Raspberry Pi, check the list of verified peripherals on the Embedded Linux wiki - **http://bit.ly/Ae6JbF**.

Members of the Raspberry Pi community are extremely active in finding and documenting which devices work and which don't, so you don't have to!

How do I set my screen resolution when using a monitor connected by VGA / DVI?

With an HDMI connection the Raspberry Pi can detect and set the resolution correctly (in most cases) for the screen that is being used. For those of us that need to connect a monitor using an HDMI to DVI cable or an HDMI to VGA convertor box the resolution needs to be set manually.

While not being as intuitive as selecting a resolution from a preferences screen, a change can be easily made by editing a config file in the Terminal. To find the correct setting for your device, look up the resolution list on the Embedded Linux wiki - **http://bit.ly/1154mFm**.

It is normally easier to find the correct resolution in the second list that shows the screen size in pixels e.g. 1024 x 768.

```
These values are valid if
hdmi_group=2 (DMT)
hdmi_mode=13   800x600   120Hz
hdmi_mode=14   848x480   60Hz
hdmi_mode=15   1024x768  43Hz
hdmi_mode=16   1024x768  60Hz
hdmi_mode=17   1024x768  70Hz
hdmi_mode=18   1024x768  75Hz
```

Make a note of the `hdmi_mode=X` setting for the resolution you wish to apply.

Within terminal enter the following command:

```
sudo nano /boot/config.txt
```

This will open the configuration file in the Nano editor to allow you to make a change. Scroll down the file to the comment

```
# uncomment to force a specific
# HDMI mode (this will force VGA)
```

Remove the # to uncomment the two lines `hdmi_group` and `hdmi_mode` and then edit them to read:

```
hdmi_group=2
hdmi_mode=xx
```

where xx is the number of the resolution mode you wish to use. Following on from my example above, for a 1024 x 768 resolution, I would use the line "`hdmi_mode=16`".

In this example, the relevant section of the config.txt file would look like:

```
hdmi_force_hotplug=1

# uncomment to force a specific
# HDMI mode (this will force VGA)
hdmi_group=2
hdmi_mode=16

# uncomment to force a HDMI mode
# rather than DVI. This can make
# audio work in
# DMT (computer monitor) modes
#hdmi_drive=2
```

After saving the file in Nano (CTRL + X, Y to confirm and Enter to overwrite) and rebooting ("`sudo reboot`") your Raspberry Pi should restart with the new resolution.

Note: If you select a resolution from the first list on the website, e.g. 720p 60Hz, you must set hdmi_group to 1.

Where there are multiple resolution settings with different refresh rates, e.g. 60Hz, 75Hz etc., you might need to experiment to find the most suitable setting.

Most computer monitors support refresh rates of 60 and 70 Hz; modern televisions are generally higher.

Why does the screen lock icon in the bottom right corner of the LXDE screen not work?

The basic Raspbian Wheezy image does not come pre-loaded with a screensaver; hence the lock not functioning and you not being able to find any screen saver preferences in the application menu.

Fixing the issue is easy enough; you just need to install the LXDE screen saver from the terminal by giving the command:

```
sudo apt-get install xscreensaver
```

Once installed, find and launch the screen saver preferences from your application menu. You should be immediately prompted that the screen saver daemon is not running. Click ok to launch it now.

Within the preferences dialog you can choose which screen saver the system will use, the idle time before it starts and if/when the system requires a password to unlock. As soon as the screen saver daemon is running the screen lock icon will function correctly.

What do I have to do to get my non UK keyboard to work correctly in LXDE?

Raspi-config does a good job of easily setting the keyboard layout correctly in the console, but those with non UK keyboard layouts often complain about their layout setting not working in LXDE.

The solution to this issue is quite simple as long you know the two letter country code and layout variant (if applicable). If you do not know this information you can find it easily from your system (assuming you have selected the correct keyboard settings in Raspi-config) by using the command:

```
cat /etc/default/keyboard
```

and checking the entries `XKBLAYOUT` and `XKBVARIANT`.

```
# KEYBOARD CONFIGURATION FILE

XKBMODEL="logicda"
XKBLAYOUT="ch"
XKBVARIANT="de_nodeadkeys"
XKBOPTIONS=""
```

With this information to hand, you can edit the LXDE autostart configuration file to add the line "`@setxkbmap XX YY`" where XX is the two letter country code and YY the keyboard variant. In my case I would edit the file by using the command:

```
sudo nano
/etc/xdg/lxsession/LXDE/autostart
```

Then in the Nano editor, append the following line to the end of the file:

```
@setxkbmap ch de_nodeadkeys
```

After saving the file in Nano and rebooting the keyboard layout should be correct within LXDE.

I hope that I have given you some pointers to get started with your Raspberry Pi and the encouragement to dig into the available community resources to resolve any small issues that you have.

For those looking for additional help I can highly recommend the following three books that are available in both print and digital formats:

• Raspberry Pi User Guide by Eben Upton and Gareth Halfacree - **http://amzn.to/ZN5EIZ**

• Make:Projects - Getting Started with Raspberry Pi by Matt Richardson and Shawn Wallace - **http://amzn.to/ZN5KKI**

• Programming the Raspberry Pi: Getting Started with Python by Simon Monk - **http://amzn.to/WyPM97**

The MagPi What's On Guide

Want to keep up to date with all things Raspberry Pi in your area?

Then this section of The MagPi is for you! We aim to list Raspberry Jam events in your area, providing you with a Raspberry Pi calendar for the month ahead.

Are you in charge of running a Raspberry Pi event? Want to publicise it?
Email us at: editor@themagpi.com

Malawi Raspberry Jam

When: Saturday 4th May 2013 @ 9:00am
Where: Bishop Mackenzie International School, Lilongwe

This event will run from 9:00am until 12 noon. Further information is available at http://raspberryjam.org.uk/event/raspberry-jam-malawi/

Swansea Raspberry Jam Session

When: Tuesday 7th May 2013 @ 6.00pm
Where: Giakonda IT, 3 Humphrey St, Swansea SA1 6BG

The meeting will run from 6:00pm until 8:00pm. Further information is available at http://www.giakonda.org.uk/raspberry-pi/

Tokyo Big Raspberry Jam 2013

When: Saturday 25th May 2013 @ 1:00pm
Where: IIJ Corp. 1-105 Kandajinbocho, Chiyoda, Tokyo, Japan

The event will run from 1:00pm until 6:00pm.
Further information is available at http://bigraspberryjamtokyo2013.eventbrite.com/

Norwich Raspberry Pi User Group

When: Saturday 1st June 2013 @ 12 noon
Where: Upper Mezzanine Level, Norwich Arts Centre Cafe, 51 St. Benedicts St., Norwich NR2 4PG

The meeting will run from 12 noon until 6:00pm.
Further information is available at http://norwichrpi.org

SWEETBOX II
The smallest Raspberry Pi case

Alexandre Xanthakis
Bud Aniontinder
Guest Writers

How to make a case for the Raspberry Pi in 314 steps

Like so many impatient technology junkies the world over, the wait to hold our first Raspberry Pi seemed like it would never end. But when we finally had it in our hands, our plan for it was quite different from most people.

After I saw the Raspberry Pi for the first time, I told my associate at Graspinghand about this unbelievably affordable, powerful computer, designed for the classroom. I thought it would be cool to make a proper case for it; the Sweetbox II. Designers often have great ideas that never get beyond being just an idea… but this idea was different.

It all started like a game. When our Raspberry Pi arrived we took extremely precise measurements during the summer of 2012 and built a 3D model. In September we started to play with the basic volume of the case, just like using Lego blocks. At first we had other projects running in parallel, but the Sweetbox II soon became an obsession.

Imagine a macro view of a bare Raspberry Pi – its components forming a fantastic and futuristic cityscape. We wanted to retain this image so a design ethos from the very beginning was that the case should wrap the components like a skin, creating a shape that would hug the circuit board and become one with the Pi. It's a unique design and also means the Sweetbox II is the smallest possible case for the Raspberry Pi. You can easily put it in your pocket! The small size also makes it easy to grasp for plugging and unplugging cables.

Components and lights had to remain visible so that people could still see what was going on inside. However, we didn't want the case to be completely clear as most transparent plastics have a cheap, shiny appearance. So we chose to make the Sweetbox II using a matt, translucent material.

Another unique feature is the integrated heatsinks. With a form-hugging design, adding heatsinks came logically in the design process and keeps the Pi cool even when overclocked. We chose copper for the heatsinks - the second best metallic heat conductor after silver - but we also have a Sweetbox II version with silver heatsinks.

We 3D printed the first prototypes in white opaque nylon and they were great… lovely to touch… the shape played with the light beautifully. But then a big problem arose; you can't make an injection mold with the type of mesh design that we'd used. 3D printing works with the STL (Stereo Lithography) format, but CNC (Computer Numerical Control) machines need a 3D model made of curves and surfaces to mill a mould. We had to completely reconstruct the geometry with a totally different vocabulary and try to recreate the same aesthetic result.

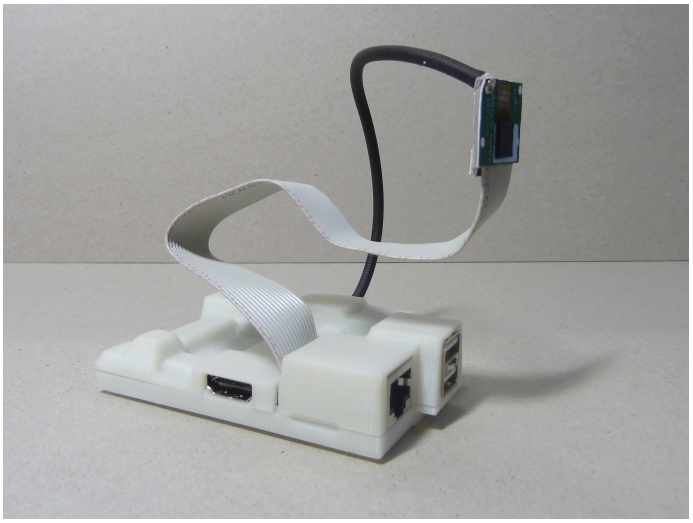
All the initial attempts that we made were disastrous, far from the shape we had at first, but after weeks of furious work we finally succeeded. The final shape was slightly different and even better than before. At last we had a good 3D model that was CNC readable.

We made eleven prototypes during the study with different materials. What was frustrating for us was that no 3D printing process seemed to offer the material translucency that we were seeking. If you want to make a translucent model with 3D printing, you first print it transparent then sand it or put a varnish on it. It means the material is only translucent on the surface rather than in the material itself. A translucent material is transparent in the first place and loaded with white powder. One of the reasons we wanted different thicknesses on the case - thicker at the base and thinner on the top - was to have a gradient effect with the tops appearing more transparent than the bases. On the final prototypes there is a homogeneous translucency that produces small differences in refractions, but we will only reach the gradient transparency that we want when we injection mould the cases.

At first we considered selling our case by 3D printing it. There are already numerous 3D printed Raspberry Pi cases available on the market, but prices remain high... too high. We have all these exciting tools available to make prototypes which are perfect for small volume manufacture. However, if you want to make a product which is available to everybody you need larger production facilities, which leads to larger costs. It's £8,000 to make a mould for the Sweetbox II plus additional costs for injection and materials!

We were searching for a way to get our project off the ground and Kickstarter was the perfect choice. Setting up a project on Kickstarter is an eye-opening process; you learn a lot about how to sell what you create. It gives you a close relationship with your audience and because what you are trying to sell is not produced yet, you can make changes to your product by following people's advice and good ideas.

The first change that we made to the prototype was to add easy access to the GPIO connector. People want to be able to experiment with their Pi and still protect the rest of the components, so we responded by making an opening with a removable rubber cap. We also added some extra space for a mini SD adapter, the pIO, which makes the overall outer dimensions even more compact! We are currently working to make the case perfect for the new Raspberry Pi camera board. We added an incision above the CSI connector to plug it in directly. A possible optional accessory is a small articulated arm attached to the Sweetbox II that will keep the camera steady.



To design a case for the Raspberry Pi is not easy. When we first held a Pi we were in awe at just how small it was. With the Sweetbox II we want to retain that awe and build on it by adding a case that showcases the Pi's size and design while providing a unique enclosure that itself will spark discussion.

An interesting consequence of the Raspberry Pi is the number of variations of cases that have been produced over the past year. We believe that all of them deserve their own exhibition; whether they are manufactured, custom made designs or homemade DIY cases. It's always fascinating to see how creative and unexpected some solutions can be. It's like music; within the same style you have an infinite spectrum of variations. Two people with the same instrument and the same constraints can each still make a unique sound.

Graspinghand is dedicated to unique productions. Our work always integrates physical properties and morphogenesis. Every project that we undertake is always special and proposes something that is genuinely unique. Our Kickstarter for Sweetbox II ends at 3:38pm GMT on 26 May 2013. For more details please visit http://kck.st/109jtN3.

All you can eat Raspberry Pi

Alan Coats
Guest Writer

DIFFICULTY : VARIOUS

In only a year the number of operating systems for the Raspberry Pi has grown tremendously. In addition to various flavours of Linux and Unix, builds of RISC OS, Plan 9 and even an Amiga compatible offering are now available. Take a look below, there is sure to be something of interest to one and all.

Default usernames and passwords are given for each OS where available. I recommend these are changed.

General

Raspbian

A free operating system based on Debian, optimized for the Raspberry Pi hardware. It comes with over 35,000 software packages.

```
Website: http://www.raspberrypi.org/downloads
Username: pi        Password: raspberry
```

Raspbian Server Edition

A stripped version of Raspbian; fewer packages installed.

```
Website: http://sirlagz.net/?p=662
Username: root      Password: root
```

Raspbian Minimal

Based on the stripped-down Raspbian image, which simply runs a Pi with sshd and a lean set of installed packages so you can start small and install the bits you need.

```
Website: http://www.pi-point.co.uk/raspbian-
minimal/
Username: pi        Password: raspberry
```

Soft-float Debian “squeeze” / “wheezy”

This image is identical to the Raspbian “wheezy” image, but uses the slower soft-float ABI. It is only intended for use with software such as the Oracle JVM which does not yet support the hard-float ABI used by Raspbian.

```
Website: http://www.raspberrypi.org/downloads
Username: pi        Password: raspberry
```

Debian 6

The Debian distro for Raspberry Pi is the Cambridge reference filesystem, which is a fully functional Debian Squeeze installation containing LXDE (desktop), Midori (browser) and development tools.
NB: Now replaced by Raspbian.

```
Website: http://www.debian.org/ports/arm/
Username: pi        Password: raspberry
```

Arch Linux ARM

Based on Arch Linux, which aims for simplicity and full control to the end user. It provides a lightweight base structure that allows you to shape the system to your needs. Please note that the Arch distribution may not be suitable for beginners.

```
Website: http://archlinuxarm.org/
Username: root      Password: root
```

Fedora Remix (4GB SD needed)

A Linux software distribution containing software packages from the Fedora Project which have been specifically written for or modified for the Raspberry Pi, and proprietary software provided by the Raspberry Pi Foundation for device access.

```
Website:
http://scotland.proximity.on.ca/raspberrypi/
Username: root      Password: fedora
```

Occidentalis (4GB SD needed)

The distro is based on "wheezy" but comes with hardware

SPI, I2C and one wire support pre-installed. It also has some things to make overall hacking easier such as sshd on startup (with key generation on first boot) and Bonjour (so you can simply ssh raspberrypi.local from any computer on the local network).

```
Website: http://learn.adafruit.com/adafruit-raspberry-pi-educational-linux-distro/
Username: pi        Password: raspberry
```

FreeBSD (4GB SD needed)

Derived from BSD, the version of UNIX® developed at the University of California, Berkeley.

```
Website:  http://www.freebsd.org
Username: root      Password: freebsdarm
```

NetBSD (4GB SD needed)

A free, fast, secure and highly portable Unix-like Open Source operating system. NetBSD is developed and supported by a large and vivid international community. Many applications are readily available through pkgsrc, the NetBSD Packages Collection.

```
Website: http://www.netbsd.org/
Username: root      Password: <none>
```

Bodhi Linux (4GB SD needed)

A Linux Distribution leveraging the fast, customizable and beautiful Enlightenment Desktop.

```
Website: http://www.bodhilinux.com
Username: pi        Password: bodhilinux
```

Risc OS (4GB SD needed)

An operating system designed in Cambridge, England by Acorn. First released in 1987, its origins can be traced back to the original team that developed the ARM microprocessor. RISC OS is owned by Castle Technology Ltd, and maintained by RISC OS Open Ltd. This version is made available free of charge to Raspberry Pi users.

```
Website: https://www.riscosopen.org/content/
```

Plan 9

Primarily developed for research purposes as the successor to UNIX by the Computing Sciences Research Centre at Bell Labs between the mid-1980s and 2002. Plan 9 continues to be used and developed by operating system researchers and hobbyists.

Plan 9 uses the 9P protocol for accessing local and remote resources as files, and provides union mounts, an improved proc file system, and native unicode support throughout. All system interfaces, including those required for networking and the user interface, are represented through the file system rather than specialized interfaces.
NB: A very experimental OS, which is loosely based on the original Unix taken to its extremes. It is only suitable for programmers at this time.

```
Website: http://plan9.bell-labs.com/plan9/
http://plan9.bell-labs.com/sources/contrib/miller/9pi.img.gz
```

Moebius

A very compact ARM HF Debian based distribution. It fits on a 1GB SD card, has auto resizing features to better adapt to your SD card size and uses Raspbian's repositories for installing everything you need. It is delivered in a small memory footprint and is ideal for working headless or for interacting with I/O devices.

```
Website: http://moebiuslinux.sourceforge.net/
Username: root      Password:  raspi
```

PiBang Linux (4GB SD needed)

PiBang, a Raspbian based distribution, is inspired by Crunchbang Linux. It comes preconfigured with many helpful scripts and pipemenus as well as a fork of Raspi-config with extra functionality. One of the heavier Raspberry Pi distributions boasting a complete package set with favourites such as Abiword, OMXPlayer, GIMP, and VLC all pre-installed.

```
Website: http://www.pibanglinux.org
Username: pi        Password:  raspberry
```

Gentoo

Gentoo Linux is a source based rolling-release distribution which emphasizes choice and flexibility. Gentoo ARM aims to be the most up to date and fastest ARM distribution available.

```
Website: http://intelminer.com/raspberrypi
Username: root      Password: pi
```

OpenWrt

OpenWrt is a highly extensible GNU/Linux distribution for embedded devices. Unlike many other distributions for these routers, OpenWrt is built from the ground up to be a full-featured, easily modifiable operating system for your router. In practice, this means that you can have all the features you need with none of the bloat, powered by a Linux kernel that's more recent than most other distributions.

```
Website:
http://wiki.openwrt.org/toh/raspberry_pi
http://www.zoobab.com/raspberry-pi-openwrt
```

Dedicated

OpenELEC

OpenELEC (short for "Open Embedded Linux Entertainment Centre") is a free and open source embedded operating system providing a complete media centre software suite that comes with a pre-configured version of XBMC and third-party add-ons with retro video game console emulators and PVR plugins. OpenELEC is an extremely small and very fast booting Linux based distribution.

```
Website: http://openelec.thestateofme.com/
Username: root      Password: openelec
```

DarkELEC
This fork of OpenELEC aims to remedy the very few flaws in its implementation and to focus 100% on the Pi, while also sticking to the upstream and incorporating its updates.

```
Website:
http://darkimmortal.com/category/raspberry-pi/
```

XBian
XBian is a small, fast and lightweight media centre distro for the Raspberry Pi, based on a minimal Raspbian image with XBMC.

```
Website: http://xbian.org/
Username: root      Password: raspberry
```

Raspbmc
Bringing XBMC to your Raspberry Pi with enough power to handle media playback, making it an ideal component in a low HTPC setup. From the developer of the Crystalbuntu Linux Distribution, which brings XBMC and 1080p decoding to the 1st generation Apple TV.

```
Website: http://www.raspbmc.com/
Username: pi        Password: raspberry
```

Chameleon
This is one for you retro gamers: a Raspbian remix from Carles Oriol that turns your Pi into a whole suitcase-full of emulated hardware, from the Spectrum to a MAME cabinet, via the Oric-1, Atari 2600, Apple II and lots of other stuff besides.

```
Website: http://chameleon.enging.com/
Username: root      Password: raspberry
Username: pi        Password: spectrum
                                (user login)
```

Elastix VOIP System (4GB SD needed)
Elastix has multiple features and functionalities related to communication services: IP Telephony, Mail Server, Fax Server, Conferences and Instant Messaging Server, among others. New features, functionalities and services are added during the development of new versions.

```
Website: http://uelastix.com/
Username: root      Password: palosanto
```

Initial web interface login:
```
Username: admin     Password: palosanto
```

Incredible Pi
An Incredible PBX platform based on Raspbian 7, featuring Asterisk 1.8 and FreePBX 2.10. It includes many of the best features of Incredible PBX including Google Voice for free calling in the U.S. & Canada.

```
Website: http://nerdvittles.com/?p=3026
Username: root      Password: raspberry
```

Raspberry Pi Thin Client
A project to support Microsoft RDC, Citrix ICA, VMWare View, OpenNX and SPICE on the Raspberry Pi.

```
Website: http://rpitc.blogspot.se/
```

Pi Point
Turn your Raspberry Pi into a Wireless Access Point

```
Website: http://www.pi-point.co.uk/
Username: pi        Password: raspberry
```

IPFire
An Open Source firewall distribution that turns the Raspberry Pi into a small router. As the Raspberry Pi computer comes with only one NIC, it works perfectly as a 3G router without plugging in additional hardware.

A generally small system that provides essential services for networks, it can be enhanced by addons which add new features to IPFire so the system can be turned into a file server and much more.

```
Website:
http://wiki.ipfire.org/en/hardware/arm/rpi
```

PwnPi (4GB SD needed)
A Linux-based penetration testing dropbox distribution. It currently has 181 network security tools pre-installed to aid the penetration tester. It is built on the Debian squeeze image from the Raspberry Pi Foundation's website and uses Xfce as the window manager.

```
Website: http://pwnpi.net/
```

ha-pi
This Debian squeeze image created to perform "pwn plug" type of attacks using Raspberry pi.

```
Website: http://sourceforge.net/p/ha-
pi/wiki/Home/
Username: root      Password: pi
```

QtonPi
Originally intended as an end-to-end platform, it enables a developer to boot straight into a controlled environment which can be deployed, via a single click, inside an associated Qt SDK.

```
Website: http://qt-project.org/wiki/Qt-
RaspberryPi
Username: root      Password: rootme
```

AROS Broadway
The AROS Research Operating System, formally the Amiga Research Operating System, is aimed at being compatible with AmigaOS at the API level. Under continuous development for several years, work has started on porting to the Raspberry Pi (note: this is flagged as "experimental").

```
Website: http://www.aros-
broadway.de/files/efika-raspberrypi.html
AROS: http://aros.sourceforge.net
```

edumacation

n. [ˌɛdjʊˈmäkeɪʃən]

1. Teaching, but fun.

2. An enlightening experience of imparting knowledge without the learner realising.

Need Raspberry Pi, Cases, Electronics, Kits or a bit of help with Physical Computing?

Have a chat with us at
edu@pimoroni.com

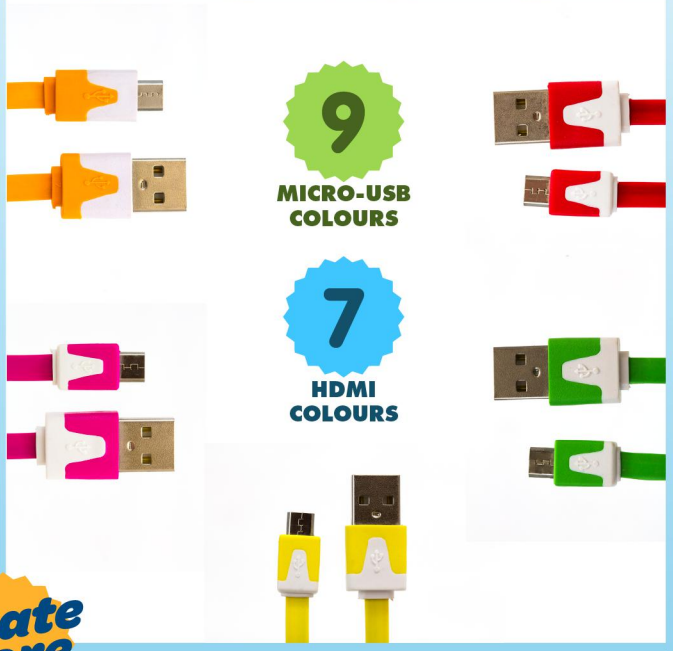
Yes, we do bulk discounts and POs and we have a lovely tame STEM Ambassador whose help is free.
We also don't mind if you point out our grammatical errors :-)

STEM AMBASSADORS
ILLUMINATING FUTURES

Noodles

HDMI 大粉 USB

http://shop.pimoroni.com



Pirate Store from PIMORONI

PI BOW

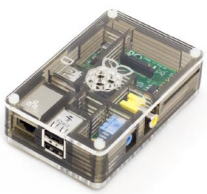
The neat little layer case for your Raspberry Pi®

Introducing the Pibow ModelA for the low-profile Model A
A hacker's delight!

Crystal

Toxic

Ninja



Adafruit

Rainbow

PIMORONI

Available From:
http://pibow.com/

adafruit · maplin · CPC · MODMYPI · MCM

A Note of Thanks from the ZX Generation

Michael Diamond

Star Writer

I was 23 when I saw a full page advertisement in the Sunday Times advertising Clive Sinclair's £70 ZX81. I had no idea what a computer was but convinced my father to fork out the money to buy me one. It arrived a few weeks later, a little plastic box, light as a feather, and a huge black power supply weighing five times as much.

I plugged it in.

Until that moment, I never really knew myself. I'd studied law at Glasgow University, for no other reason than my brother did before me. I was in the middle of my law apprenticeship when the ZX81 arrived.

From the moment I switched it on, I was hooked. I went through the manual, a page at a time, realising that each line of BASIC was infinitely variable. A little tweak here, a minor adjustment there, and the ZX would do anything I wanted. I spent months with that little thing, fiddling around, copying lines of BASIC out of magazines, and writing my own programs to play mastermind, hangman, and even multicoloured beeping music.

That was 30 years ago - when Eben Upton was just four years old.

While my family humoured me, my legal friends couldn't grasp what I was talking about; they'd laugh disdainfully at my enthusiasm about a little plastic box.

Little did any of us know that computers were about to change the world.

Captivated as I was, eventually my professional training got in the way; younger kids with computer education went on to open software houses, while I watched from the legal sidelines.

Half a lifetime later, and the Raspberry Pi is born. I have my own grown up kids now, and computers to them are like pen and paper were to me. When I told them about the Raspberry Pi, they didn't get it. I bought one anyway - the same urge that brought me the ZX81 years ago.

It arrived six weeks ago - and I've stepped back in time.

In those six weeks I've learned so much. Until then, I had no idea what Linux was, or GPIO ports, or HDMI, SSH, or how transistors work. I didn't know what a breadboard was, or a web server, or a breakout board, or a 555 timer, or php. I didn't know what a static IP address was, or Dynamic DNS or how to forward a port in my router. In a month and a half, I've learned all of these things and more. At first I lit a single LED. Then for £2, I bought a relay and progressed to switching on a reel of them. Then I did it by remote control - in the house - from my smart-phone. The next step was obvious - hook it to the internet.

I showed the interface to my office staff, telling them, “when I click this switch, a light goes on in my house.”

“How do you know?” came the intrigued reply.

“Hmm,” I thought. “How do I show them?”

So this weekend I hooked up a webcam to my Raspberry Pi. Now, when I click the switch, through my browser, I can show them the LEDs come on. When it crashed, I reset the whole thing from afar - amazing even myself.

If someone had suggested to me two months ago that I’d be capable to doing something like that in just six effortless weeks, I’d have said she’s crazy. Yet here I am.

I’m sure I’m slower than the younger folk, but at 54, with no formal technical experience, I’m filled with satisfaction and gratitude for the thousands of people out there who are sharing their knowledge on YouTube and blogs. I can't contribute much technical knowledge to the community, but I can express the gratitude of the ZX81 generation to Eben and his team for re-igniting our Sinclair days, three decades later.

Many, many thanks for your enthusiasm and good work. I am sure there are thousands of people out there just like me, tinkering, hacking and crying out with excitement when their little project bursts into life.

Michael Diamond

KICKSTARTER

Thank you again to everyone who has supported our Kickstarter campaign to bring The MagPi to the masses in hard copy form. As unpaid volunteers this was a massive challenge for us and one we are very proud to have accomplished. I would like to take this opportunity to once again apologise for the long waiting times between ordering and us dispatching; as with any challenge, we have experienced hurdles on the way to meeting our goal plus we significantly under-estimated the required effort.

Additional time has been taken to convert issues one, two and three from our original format into Scribus. This meant effectively starting from scratch; a massive task. Bleeds had to be adjusted on every page of all issues. On top of this we faced and overcame problems at our printers, issues with our proofs and very narrow timelines to meet. Looking back at the effort, it’s amazing we have got this far!

To minimise delays, we got the hardware orders from pledges out to our supporters. This resulted in extra shipping costs, but as a team we felt it was worth it to give you something to play with and reflect our appreciation of your patience.

We are pleased to write that almost all 256 pages have been revisited and updated, providing you with current information and URLs, in The MagPi style and layout that we used in later issues. This effectively translates to creating 12 magazines in four months, three per month including this issue you are reading. We have been very busy!

With that said, I am very pleased to say that as we speak, our printers and distributor are working hard behind the scenes to package up your bundle. Expect yours in the post soon!

Many thanks once again, we appreciate your patience. We hope you continue your support of the largest and only community led Raspberry Pi magazine as we continue into the future.

Ash Stone

2 - Printing from Python

DIFFICULTY : INTERMEDIATE

Stewart Watkiss

Guest Writer

Last month, we added CUPS printing onto the Raspberry Pi. Now that you have got your printer up and working we can look at how to print from within our own applications. We are going to be using the Python programming language, but we will also take a look at straightforward command line printing which can also be used for printing from other programming languages.

First, we will look at printing from the command-line. This can be useful within shell scripts or by making a system call within any programming language. The `lp` or `lpr` commands are used for command-line printing. The names are an abbreviation of line-printer, although they are seldom used for printing to a true line-printer these days. Both commands work in a similar way, but use different options. This is historical from the early days of UNIX. Both `lp` and `lpr` are available on Linux so you can use whichever you prefer.

The command to print a file to the default printer is:

```
lp <filename>
```

You can select a different printer or specify only certain pages using different command-line options. You can find details at:
http://www.cups.org/documentation.php/options.html
or by looking at the help page on your own CUPS configuration page which we used in the previous article.

Unlike the original commands which could only handle text files, the lp and lpr commands installed through CUPS can handle other formats as well. When you use the commands with a PDF, postscript or image file then CUPS will convert the files using the printer drivers. We will be using this feature later when we print a PDF file from a Python script.

In some programming languages and GUI toolkits (see box) it is possible to generate printer output using similar instructions to how you generate graphical content for the screen. This is the case in PyGTK,PyQt and WxPython; but whilst the GTK and Qt toolkits are commonly used with other programming languages they are less used and less well documented on Python. WxPython can be useful if you are using Wx toolkit already, but the most common graphical toolkit on Python is Tkinter, which has very limited support for printing.

GUI toolkit

A GUI toolkit provides a standard set of buttons, frames and other components to make creating graphical programs easier and with a consistent look and feel. Some include additional printer libraries that make it easy to convert the graphical screen output to a format ready for printing.

Instead of using toolkit support I have chosen to generate a pdf file using xhtml2pdf which is sent to the printer using a CUPS library called PyCups. This is easy to use and has the advantage that it doesn't need to be used in a graphical environment so can be used from within a command-line program or a daemon application running in the background.

First, we will install some additional packages. The python-cups library provides the module so that Python can communicate with the CUPS printers, python-pip will make installing xhtml2pdf easier and the python-dev libraries are needed to allow pip to install some of the libraries used by

xhtml2pdf. Install these using the normal apt-get install:

```
sudo apt-get install python-cups python-pip
python-dev
```

Then install xhtml2pdf using pip:

```
sudo pip install xhtml2pdf
```

This last stage will take a while to run as it will download and compile some additional libraries.

You can now use xhtml2pdf on the command-line to convert from a HTML page to a PDF file. You could try using a saved web pages, but in many cases it will only partially work due to the common use of Javascript which isn't rendered by xhtml2pdf. Instead I recommend creating the pages using standard HTML markup.

Below is a Python script which will list the available printers and then print to the first printer in the list. This is an example created to show how you can add printing capability to your own program. If you are including this into your own application then you should add additional error checking as well as a way for the user to select the appropriate printer (if multiple printers are installed).

Line 4 imports the cups library and line 5 the xhtml2pdf library (note that it is imported as pisa which is the former name for the library and is how we will refer to it later).

Lines 11 to 14 generate some basic HTML formatted content. Whilst you could include the usual HTML headers and head section, which you would normally include in a html page, I have just added the content of the body section which is sufficient for xhtml2pdf. We then use CreatePDF on line 16 to convert that to a file and save it to the file specified in variable “filename”.
I have included a very basic check for PDF errors at this point as invalid HTML could cause this to fail. This is only a generic error message which you may like to improve on in your own programs.

Line 19 is very important. The file needs to be closed so that the write completes and it's possible to read the file back in. Without this then nothing will print, and in my case I spent a lot of time tearing my hair out trying to work out what was going wrong.

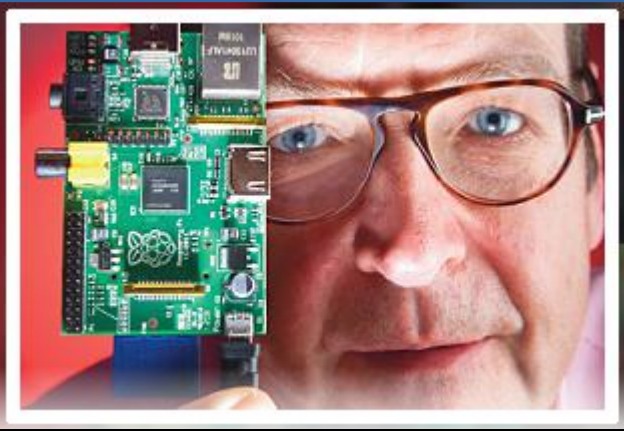
We create a connection to CUPS and use the getPrinters command to get a list of the available printers. I've deliberately included the print command on line 27 which is the standard Python command to print a line of text to stdout (screen). It is nothing to do with sending anything to the printer, so don't let that confuse you.

Finally on lines 29 and 30 we select the first printer and print the file using the printFile command.

When creating the script you need to create a name that doesn't conflict with any of the names of the library files imported. For instance do not call the script “cups” as that will conflict with the cups library. I used cupsprint.py during my testing.

This has shown one of the ways that you can add printing to your own Python application. This method has been chosen as it's easy to implement and can be used for command-line and server applications as well as graphical applications. So over to you; what projects do you have that need to be able to print?

```
1. #!/usr/bin/env python
2. # Print a file to the printer
3.
4. import cups
5. from xhtml2pdf import pisa
6.
7. def main():
8.   # Filename for temp file
9.   filename = "/home/pi/print.pdf"
10.
11.  # generate content
12.  xhtml = "<h1>Test print</h1>\n"
13.  xhtml += "<h2>This is printed from
     within a Python application</h2>\n"
14.  xhtml += "<p style=\"color:red;\">
     Coloured red using css</p>\n"
15.
16.  pdf = pisa.CreatePDF(xhtml,
     file(filename, "w"))
17.  if not pdf.err:
18.    # Close PDF file - otherwise we can't
       read it
19.    pdf.dest.close()
20.
21.    # print the file using cups
22.    conn = cups.Connection()
23.    # Get a list of all printers
24.    printers = conn.getPrinters()
25.    for printer in printers:
26.      # Print name of printers to stdout
         (screen)
27.      print printer,
         printers[printer]["device-uri"]
28.    # get first printer from printer list
29.    printer_name = printers.keys()[0]
30.    conn.printFile(printer_name, filename,
       "Python_Status_print", {})
31.  else:
32.    print "Unable to create pdf file"
33.
34. if __name__=="__main__":
35.   main()
```



INTERVIEW

Ian McAlpine
MagPi Writer

Pete Lomas shares his thoughts on Raspberry Pi design challenges and the future.

Pete Lomas is one of the trustees of the Raspberry Pi Foundation. He is also the man responsible for the final hardware design of the Raspberry Pi. The MagPi met up with Pete at the Vancouver Hack Space, while he took time out from a recent vacation to Canada.

[MagPi] How did you first get involved with the Raspberry Pi?

[Pete] It was a chance meeting with Alan Mycroft at Imperial College where my company, Norcott Engineering, had been commissioned to do some work. I was discussing with Alan that graduates interviewing at Norcott were not "engineer-ready" - they were "half-baked" because they didn't have the practical skills. Alan agreed that they had observed the same problem with students coming to Cambridge University. However, he then explained that a chap at Cambridge called Eben Upton had the idea of building a small, single board computer that they could give to undergraduates before the start of their first semester. The implication was if you couldn't do anything useful with it, don't come back! I thought this was brilliant because the introduction of the PC started to package computers and remove the ability to "tinker" that was rife during the home computer boom of the 80's. During the last ten years especially, consumer electronic products have become so polished that kids today are becoming pure consumers and not creators. The skill of "tinkering" is lost. I recognised that Eben wanted to encourage "tinkering" and so I got involved.

[MagPi] I was 14 years old in 1982 and was fortunate that my parents bought me a BBC Micro Model A. It didn't work! The case was only held on by four screws and there was a large slot at the back to easily peek inside. I removed the case, spotted a badly seated IC and also discovered the TV antenna cable had not been soldered. Thirty minutes later and it was working. The BBC Micro Model A cost £299 back in 1982. Imagine trying to open up or repair an equivalent priced computer today! Last year I restored that very same computer back to full working order, while waiting for my Pi to be delivered.

[Pete] Precisely. Back then you were encouraged to "tinker". Today not only do manufacturers not want you to "tinker" but the financial repercussion if something goes wrong is significant. Price is one of the most important aspects of the Raspberry Pi. It had to be cheap enough that it could be given to each child. It's theirs to do with whatever they want without any fear... and if they do break it, it's not the end of the world.

[MagPi] Now that you have brought up the topic of breaking the Raspberry Pi, some of our readers have asked if you could relocate the large capacitor beside the micro USB connector in a future revision?

This will stop it from snapping off if your finger slips while trying to insert or remove the power cable!

[Pete] The Raspberry Pi was designed using the best engineering practices. C6 is not essential if you have a good power supply. Ideally it should be fitted, but don't worry if you have broken it off. Interestingly, I've used hundreds of Raspberry Pis and never once broken off that capacitor. People tend to hold the Raspberry Pi tentatively; they are afraid it might break and that is what causes fingers to slip. Instead just grab the Raspberry Pi firmly and yank out the power cable. Look carefully at the micro USB power socket. It is mechanically attached to the circuit board. We went to great lengths to ensure that the power socket will not snap off.

[MagPi] Would the use of USB fuses on the revision 1 board be an example of good engineering practice?

[Pete] With the benefit of hindsight, the USB fuses will forever be one of those points of regret. Best engineering can still be a compromise on price / performance (except, of course, when lives are at risk). We intended to limit the current to these devices based on the testing we had done with mice and keyboards. However, we did not fully appreciate the inventiveness of the people that started to use the Raspberry Pi. We had focused on the strategic goal of the education market and somewhat forgot the hobbyist market that wanted wi-fi dongles, bluetooth and USB hard drives. Of course the problem was resolved with the revision 2 board but that is always the problem with design; you can be over cautious and it will bite you at one end, or under cautious and it will bite you at the other end!

[MagPi] What were the biggest hurdles with designing the final version of the Raspberry Pi?

[Pete] Keeping the cost of the Model B at $35 was paramount. That barrier simply could not be broken so I had to be ruthless about every single component. At one point I wanted to remove the ethernet connector but Eben, Jack and Rob all grabbed me by the throat and strongly encouraged me to leave it on! The focus on cost also meant making the Raspberry Pi as small as possible but also within the constraints of the physical connectors.

At a trustee meeting someone mentioned it should be credit card sized. I drew a credit card sized outline on the CAD system and discovered I could make everything fit!

[MagPi] Using an SD card for the operating system and for storage has advantages and disadvantages. What was the rationale for using SD cards?

[Pete] SD cards are commonplace, robust and inexpensive. They are frequently used in digital cameras so it is quite likely you already have some. They are also quick and easy to flash. If something goes badly wrong, simply reflash the SD card and you are back with a new Raspberry Pi. You can also have different SD cards for different uses; a card for school, a card for home projects, a card for RISCOS, a card for XBMC... and each one can completely transform your Raspberry Pi.

[MagPi] If you could change anything on the Raspberry Pi, what would it be and why?

[Pete] If I had known just how rubbish some 5V USB power supplies are, I would have taken measures to ensure a clean 5V is delivered to the various systems in the Raspberry Pi. Thanks to the community however, that problem is largely behind us as people now know to only use a quality power supply.

[MagPi] Does the education version of the Raspberry

Pi exist and if so will there be any difference between this version and the current version (Revision 2) of the Raspberry Pi?

[Pete] The education version is in development - we are actively working on it at the moment and it will have a case. There are some things that we plan to address. For example, with an install base of 1,000,000+ units many people use HDMI-VGA converters to repurpose old VGA monitors. A year ago these were expensive, but the demand from Raspberry Pi users is such that these now cost less than $10! I want to adjust the power to the HDMI connector so that these converters can be reliably used.

[MagPi] The Raspberry Pi branding pays homage to the BBC Microcomputer Model A and Model B. Are there plans for a Raspberry Pi Model B+ or Master?

[Pete] We have doubled the RAM on the Model B to 512MB so that is the equivalent of the BBC Micro Model B+! The Raspberry Pi Foundation does not have the same mandate as a commercial business. We don't need to release new versions or new products every quarter to please share holders. We do not want to disenfranchise early adopters of the Raspberry Pi. We may revisit it again in a couple of years as technology changes, but not yet. The Pi would not be as successful as it is were it not for the community. We want the community to continue to grow with the current hardware.

[MagPi] Another success story for the Raspberry Pi is that despite its low cost, it is being manufactured and assembled in Wales, in the UK; British designed and British built.

[Pete] People have no idea of the lengths that Sony UK have gone to get the manufacture error rates and costs down as low as they are now. They are world class and the attention to detail is mind boggling. But even more impressive is that Sony UK have completely taken onboard the Raspberry Pi philosophy. They host factory tours for bus loads of kids and they don't skimp... they do everything to ensure that everyone has a fantastic time. They, like us, want to educate kids that engineering is exciting, rewarding and fun.

[MagPi] What can you tell us about the new camera?

[Pete] The camera is currently going through the production process and people in Cambridge are optimising the drivers to get the very best quality. But even when you run as hard as you can, everything still takes time. If you don't take the time to do things right then you just make rubbish! The camera should be available in May 2013 and I'm sure that, once in the hands of the Pi community, we can look forward to some inventive uses for this new hardware.

[MagPi] What about plans for other add-ons, for example displays using the DSI connector?

[Pete] We do have the intention to make displays available for the DSI connector, but it's not on our priority list. The camera integrated onto the Raspberry Pi is a priority because it broadens the horizons of what you can do with it. But a plethora of different displays and screen sizes are already available today that use either the HDMI, composite video, GPIO or I^2C interface, so there is less urgency to develop a dedicated DSI display.

[MagPi] Why was the decision made to create the Raspberry Pi when other low cost devices such as the Arduino were already available?

[Pete] If the Arduino could run Linux out of the box, had an HDMI connector and a GPU, then we probably would have used it. But it doesn't... it's a very different product that allows you to do very intricate, time-dependent things with its I/O that you cannot do with a Raspberry Pi. They complement each other; they do not compete.

[MagPi] Indeed, we have published articles in issues 7 and 8 showing how to use the Raspberry Pi and Arduino together.

[Pete] But remember we needed something to get kids excited, so we had to have good video and sound. You can configure the Raspberry Pi to start in whatever programming environment you want when you switch it on. When you turned on a BBC Micro or ZX Spectrum and saw the blinking cursor, you knew the computer was waiting for you to tell it to do something. You were the master... and this can be

very empowering for kids. You show them Scratch and the cat moving around the screen then come back 5 minutes later and the cat is doing all sorts of interactions. All the code building blocks are down the side of the screen and kids think, "I wonder what that does?" and they try it out and learn. The Raspberry Pi is small and its circuitry is visible so kids know there is no magic. The Raspberry Pi will only do what they tell it to do, so the ownership value is very high.

[MagPi] The physical computing side of the Raspberry Pi is very exciting, not just for me as an engineer, but even my two pre-teen daughters are very impressed that they can use Scratch or Python to turn on LEDs or interact with the Raspberry Pi using physical switches.

[Pete] Look at the articles we publish on the raspberrypi.org website. Fifty percent have something to do with connecting the Raspberry Pi to some other hardware. That means we are hitting a niche that no iSomething will ever meet.

[MagPi] One year on, what projects have impressed you the most?

[Pete] The Raspberry Pi inspires the maker in all of us. I immediately think of projects like the "Brew Pi" and of course Dave Akerman's "Pi In The Sky". But even more impressive is watching what kids are doing with the Pi. If I'm slogging through some work or feeling down, watching videos of kids like Amy Mathers at the Raspberry Jamboree or the TEDx video of the girls from Charlotte Latin School and seeing their passion and enthusiasm really gives me a boost to get on and continue working.

Hacker Spaces

Pete was presenting at the Vancouver Hack Space (VHS) to a packed room of over 40 makers. What an amazing place with all manner of tools, components and equipment including several 3D printers and a laser cutter. The MagPi spoke to several makers there, including Richard Sim and Loial (Matthew Peters). Having built a couple of "off-the-shelf" 3D printers, Loial thought he could design and build a better one... so he did! The TS300 is a 3D motion platform that prints twice as fast as most other 3D printers and it's powered by a Raspberry Pi! The TS300 will soon be available on Kickstarter. Visit http://www.tesseractindustrial.com for more details.

Commenting on the Vancouver Hack Space, Loial explained, "Having access to tools and equipment means people can create projects that they would normally not be able to do. But more than the equipment, the Hack Space brings together very creative people with a 'can do' attitude from all walks of life. It is their collective knowledge which has certainly helped me bring my project to reality."

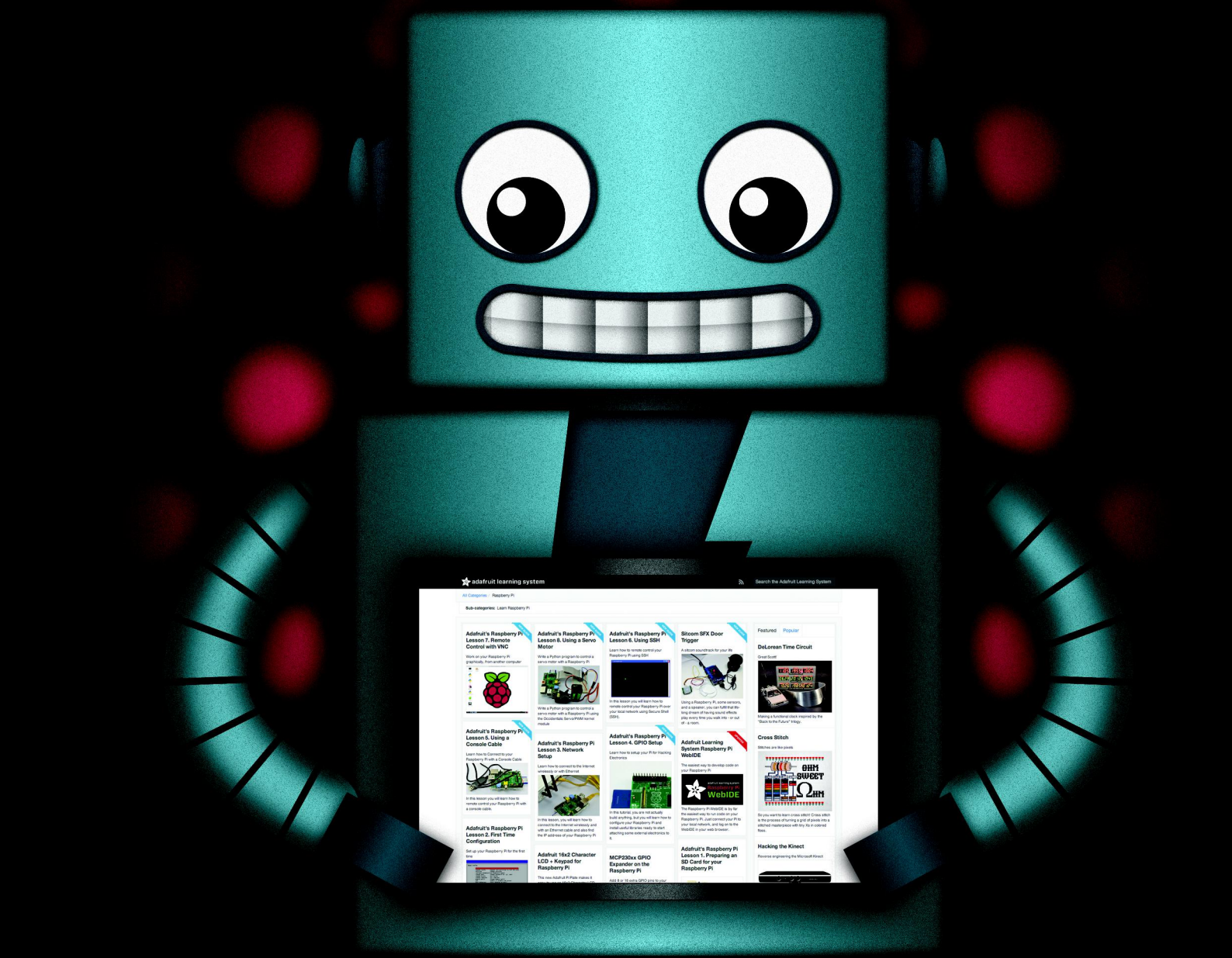
As we concluded the interview Pete quipped, "When I finally pass on, my epitaph will not say 'Peter Lomas - RIP', it will say 'Peter Lomas - RPi'"!

On behalf of The MagPi I would like to thank Pete Lomas for taking time out from his vacation to talk to us. I would also like to thank the VHS for their kind hospitality. They regularly host Raspberry Pi evenings. To find out about upcoming VHS meetings, visit http://vancouver.hackspace.ca. To find a hack space near you, visit http://hackerspaces.org and click on the 'Active hackerspaces' link.

adafruit learning system

Raspberry Pi WebIDE

BUILD AMAZING THINGS & LEARN HOW TO PROGRAM WITH THE RASPBERRY PI


adafruit learning system
Search the Adafruit Learning System
Adafruit's Raspberry Pi Lesson 7. Remote Control with VNC
Adafruit's Raspberry Pi Lesson 8. Using a Servo Motor
Adafruit's Raspberry Pi Lesson 6. Using SSH
Sitcom SFX Door Trigger
Featured
Popular
DeLorean Time Circuit
Adafruit's Raspberry Pi Lesson 5. Using a Console Cable
Adafruit's Raspberry Pi Lesson 3. Network Setup
Adafruit's Raspberry Pi Lesson 4. GPIO Setup
Adafruit Learning System Raspberry Pi WebIDE
Cross Stitch
Adafruit's Raspberry Pi Lesson 2. First Time Configuration
Adafruit 16x2 Character LCD + Keypad for Raspberry Pi
MCP230xx GPIO Expander on the Raspberry Pi
Adafruit's Raspberry Pi Lesson 1. Preparing an SD Card for your Raspberry Pi
Hacking the Kinect

adafruit

HTTP://LEARN.ADAFRUIT.COM/CATEGORY/RASPBERRY-PI

Rock solid protection for your gadget.

Machined Aircraft Aluminum
Anodized Color Options
Security Features
Passive Cooling
Affordable

PROTO ARMOUR™

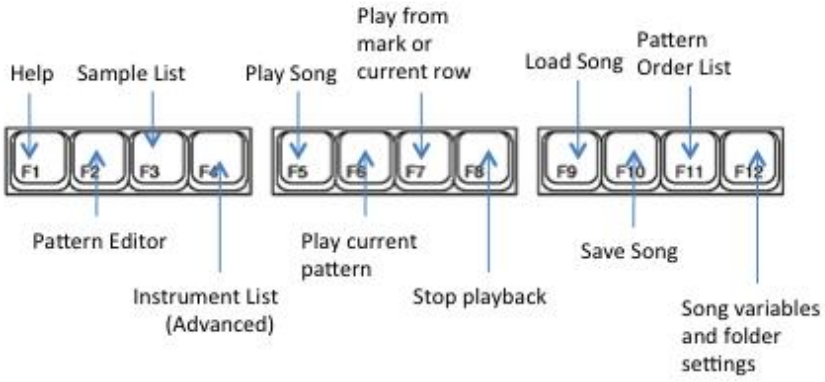
WWW.PROTOARMOUR.COM

Schism Tracker

David Honess
Guest Writer

Creating music using the Raspberry Pi - Part 2

DIFFICULTY : BEGINNER



Welcome back folks. If you missed part 1 of this article please go and check it out in The MagPi issue #2. Above is a quick reference sheet for the Schism Tracker function keys that you can keep referring back to while you read this article. It can be helpful to print this out and put it on your wall.

What we're going to do

By the end of this we should have programmed the melody for DJ Quicksilver's Bellissima. This is a famous dance tune that came out in 1997. You've probably heard it before, but you just don't know it by name. You can Google it if you want to. The tune has quite a simplistic melody but a catchy one and it is ideal for getting to grips with Schism Tracker.

Setting up

I'm going to assume we're using Raspbian for this. Firstly we need to install the Schism Tracker program. You can either use the Pi Store to install it, you will find it under Apps, or you can use the command

```
sudo apt-get install schism
```

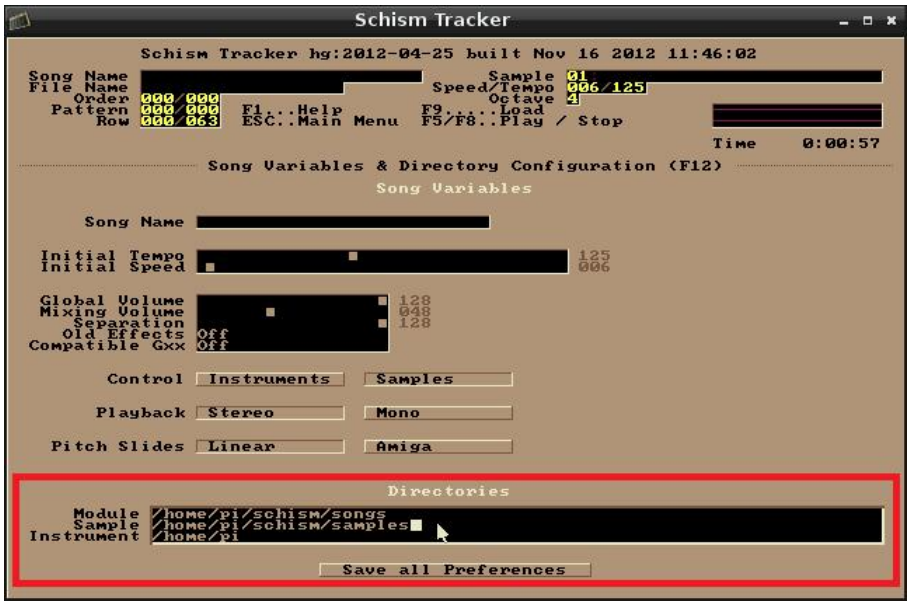
If you install it via the Pi Store you can launch the program via the Pi Store > My Library area or, if you used apt-get, you can find Schism Tracker listed in the LXDE desktop launcher menu under Other.

First we need to create some folders to store our files. We need one folder to save songs files and another for sound effect files (instrument sounds). Start the LXDE desktop if you haven't already (`startx`). Open the File Manager. It should display your Home folder.

From the top menu select; File > Create New > Folder. Name the folder `schism`.

Open this folder and inside it create two more folders, one called `songs` and another called `samples`.

Run Schism Tracker and press the F12 key. At the bottom part of the screen you should see an area labelled 'Directories'. There should be three rows; Module, Sample and Instrument. Don't worry about Instrument for now. Change module so that it says `/home/pi/schism/songs` and change Sample so that it says `/home/pi/schism/samples`. Once you're done click Save all Preferences.

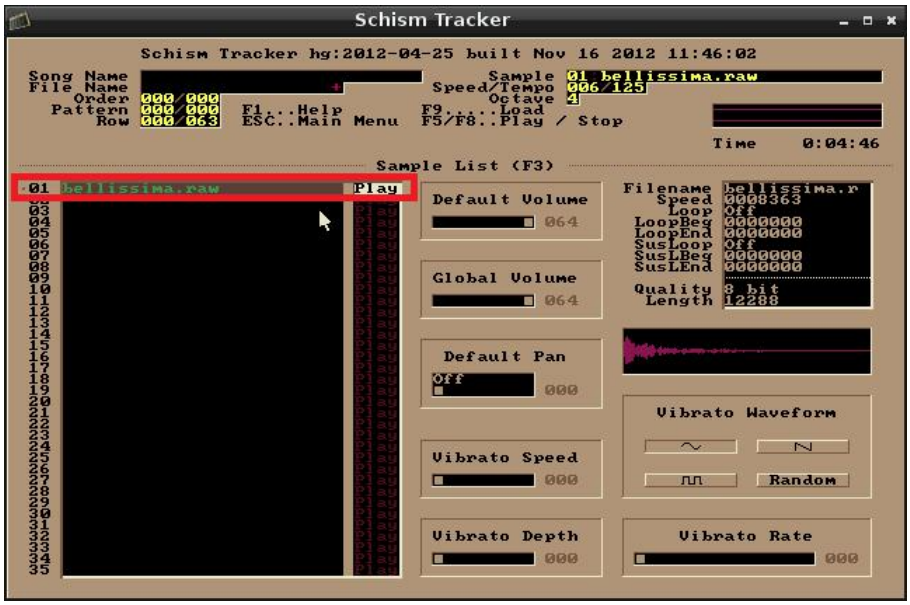


Downloading a sample

We have a prepared sample ready to use for the main melody. You can either use Midori or wget from a command prompt to download it: https://dl.dropbox.com/u/14125489/RaspberryPi/Schism/bellissima.raw. Save this file into the samples folder that we created before (or save it anywhere and then move it to the samples folder).

Loading the sample

Go back to Schism Tracker. Don't load any songs - just leave it blank. Press the F3 key. This is the sample list page. Don't worry about all the stuff on the right hand side. Look at the list on the left, this will be where each sound effect or vocal will be loaded in your song. One can be loaded for each row. Select any row you like and press the Enter key.



You should now see a row showing belissima.raw. Select it and press Enter again. You should now be back on the sample list page. Keep the belissima sample selected (put the cursor over Play) and run your finger left and right between Q and P on your keyboard. You'll see this is like the notes of a piano. Some of the number keys on the row above represent the black keys too.

Try this

Stay on the sample list but press these keys with the belissima sample selected (keep the cursor over Play).

```
6 6 6 e e e y y 5 5 5 u u y 9 y
```

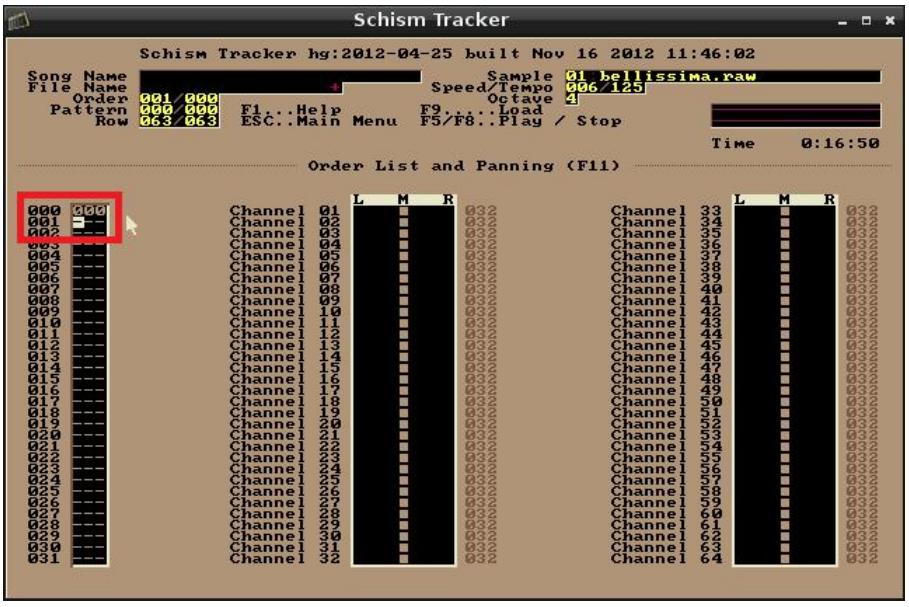
You should be able to make out the melody of the tune! Now we have to program it into a sheet of music, known as a pattern, so that the melody is stored somewhere.

Programming the melody

Press the F2 key to go to the pattern editor screen. You are now looking at a pattern number 000 (think of it like a sheet of music). You'll see across the top Channel 01, Channel 02 etc and down the left hand side numbers running from 000 to 063 (so 64 rows in total).The way this works is that we select a sample from the sample list (F3) go to the pattern editor screen (F2) and program notes into the available rows of a channel. Ideally you should always use a separate channel for each sample but you don't have to. When the computer plays the pattern it moves down the screen playing all the notes that occur on each row (across all channels). The effect is that sounds from each channel are played together at the same time (mixed together) to form the overall sound of the song. Let's put our melody into Channel 01. The cursor (a small white square) should already be inside the channel, you can move it around using the cursor keys or with the mouse. Start at row 000 and on every other row type in a note from the melody sequence:-

```
6 6 6 e e e y y 5 5 5 u u y 9 y
```

You should finish up on row 030 (see below). Now press the F6 key to play back the pattern. Watch how the white highlight in the row numbers moves down the left side. Press F8 to stop.

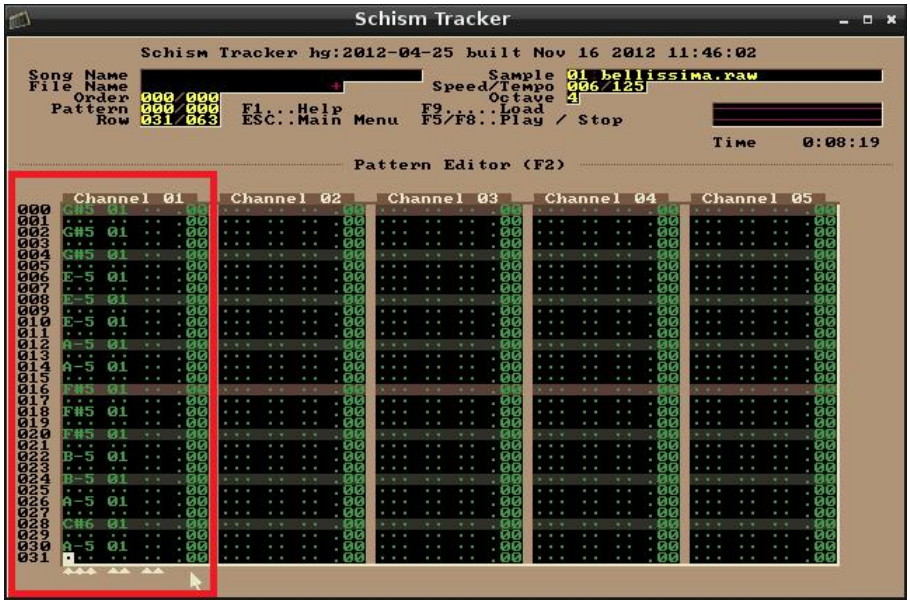


You will notice that there is a gap of silence before the melody starts up again. This is because we have not put any notes below row 030 in the pattern. If you now go to row 032 and repeat the same sequence of notes finishing on row 062 the melody will now loop continuously when you press F6. Press F8 again to stop.

Congratulations you have just programmed your first pattern! Now we need to make this pattern part of the actual song. You'll notice that if you now press F5 to play the song, like we were doing in the previous section, nothing happens. This is because the pattern order list has not been set yet. Press the F11 key for this.

Once you're on the pattern order list screen don't worry about the middle and right hand columns. The left hand column is where the pattern order is kept.

Go to the first row and enter 000 as shown above (this is the number of the pattern you have just programmed). Now if you press F5 you'll hear the music (just like when you play a normal song file). Press F8 to stop the playback and go back to the pattern order list again (F11). On the second row enter 001. Now press F5 again. You'll hear the melody play and then it will go silent for a bit. This is because we have not programmed anything into pattern 001 yet.



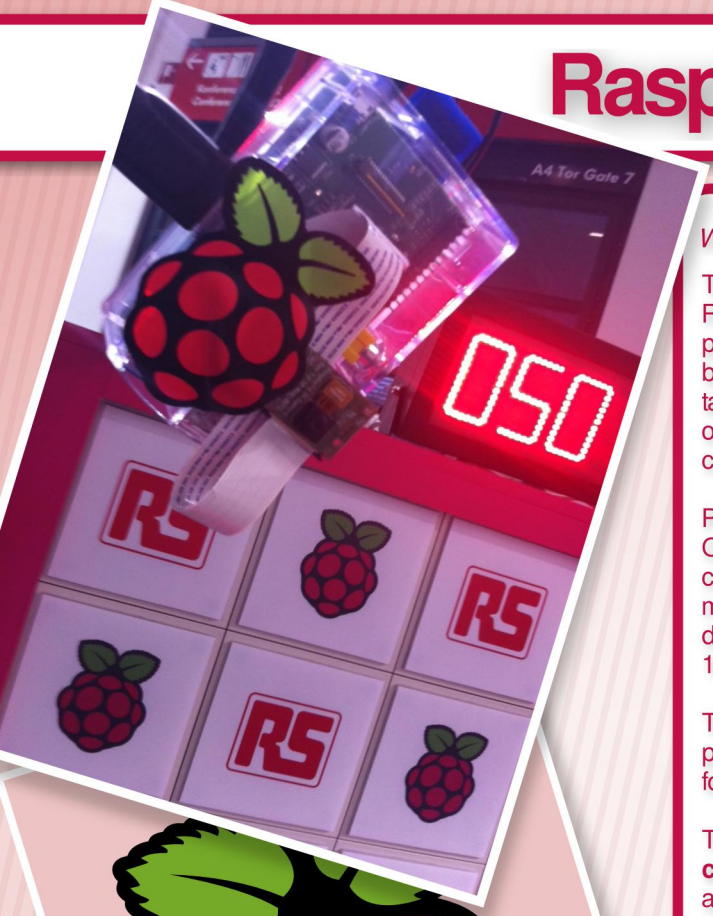
Copy and paste

Press F2 again to go back to the pattern editor. You can use the + and - keys to switch between patterns. Notice the pattern number changes at the top of the screen where it says Song Name, File Name, Order, Pattern and Row.

Before you start worrying that you have to type in all the notes again, there is a copy and paste function. This is ideal if you want the next pattern in the song to be the same but with one extra sound for instance. Go back to pattern 000 and put the cursor at the top of Channel 01. Hold down the Shift key and use the cursor/arrow keys to select the entire column (from rows 000 to 063). Press Alt – C to copy the notes into memory. Now press the + key to go to pattern 001. Place the cursor at the top of Channel 001 and press Alt – P to paste.

We need another sample to make it sound like something is happening in the song. Use Midori or wget to download this sample: https://dl.dropbox.com/u/14125489/RaspberryPi/Schism/kickdrum.raw. Save it into the `samples` folder like before. This is a meaty sounding kick drum which will work quite well with the melody we already have. In Schism Tracker press F3 to go to the sample list, select a blank row and press Enter. Select the kickdrum.raw file and press Enter again.

Raspberry Pi Camera Board



Writes Glenn Jarrett, Global Head of Product Marketing at RS Components

There was a great deal of interest when we first demonstrated the prototype Raspberry Pi camera module at Electronica in Munich last November, and the final product has been eagerly awaited. So we are delighted to announce that the first batch of camera modules is now ready and will be winging its way to us very shortly. I take my hat off to the team at the Raspberry Pi Foundation who have been working flat out to fine-tune the hardware and driver to get it into production and into our distribution centres as quickly as possible.

Priced at just $25, the module contains a 5-megapixel CMOS image sensor from OmniVision mounted on a 25 x 20 x 9mm PCB, delivering still image capture and the capability for high frame rate 720p/60 high-definition (HD) video capture. The camera module interfaces to the Raspberry Pi via the existing camera connector, using CSI for data and I2C for control. At the time of launch it will allow users to record 720p and 1080p at 30 frames per second in H264 video format.

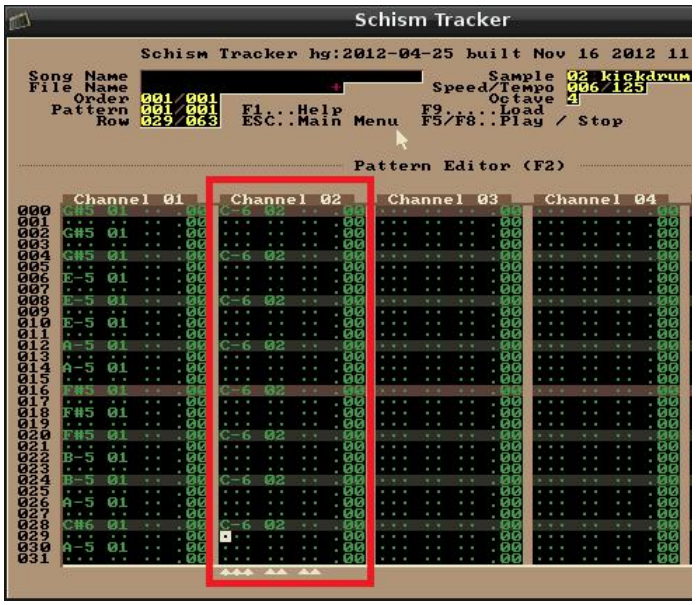
The final picture quality of the product is excellent thanks to the superior pixel performance of the image sensor it uses. This gives customers a very low cost option for adding video input to their projects for anything from robotics to home automation.

The Raspberry Pi camera module will be available to purchase from **www.rs-components.com/raspberrypi** We will let you know as soon as we have stock available so look out for announcements from RS in the press and on Twitter @RSElectronics over the next few weeks. To read more about how to integrate the camera module in your design or to find ideas and tips for using it, why not visit www.designspark.com, the RS online design community, where you will find a dedicated Raspberry Pi design centre packed with useful information and tutorials.

www.rs-components.com/raspberrypi

You can now test it out with the Q to P keys again. In my opinion the I key sounds best.

With the kick drum sample selected switch back to the pattern editor, F2, and put the cursor at the top of Channel 02. Use the I key to place kick drum beats every 4 rows. Make them follow the highlighted lines as shown below. Go all the way down to line 60.



If you now press F5 to play the entire song, you'll hear the melody loop play twice before the kick drum comes in. Cool hey?

Saving to disk

Okay let's save the song to disk now. Press the F10 key, enter a file name and press Enter.

Congratulations, you have just made your first module! It might be time to turn the volume up, marvel at what you have just done and annoy your parents a bit!

The MagPi's HUGE Birthday Competition

Over £1,000 worth of prizes to be won!

It has now been a year since the first issue of The MagPi was released. The year has gone by very quickly and we have been very busy... but don't worry, we left enough time to organise this huge competition to celebrate our 1st Birthday in style! Think of it as a small way for us to say a big thank you for all of your support, patience and understanding over the last 12 months. To be in with a chance of winning one of these fantastic prizes, all you have to do is answer the multiple choice questions below (and optional text questions) at www.themagpi.com/birthday. All correct entries will be put in to a prize draw, with the first 15 drawn winning a prize in the order listed. Winners will be notified by email and the winning names will be posted on The MagPi website.

Full terms are available at: www.themagpi.com/birthday

QUESTIONS

Q1 How many times on average each month is The MagPi downloaded or read online?
- a) 1,500 times
- b) 16,000 times
- c) 175,000 times

Q2 In how many of the 204 countries of the world have people NOT read The MagPi (based on our Google Analytics site hit data as of 1st April 2013)?
- a) 107 countries
- b) 52 countries
- c) 8 countries

Q3 How many pages of the The MagPi have been created from Issue 1 through to issue 12 inclusive?
- a) 256 pages
- b) 384 pages
- c) 404 pages

Q4 In which issue is there an article about Arch Linux?
- a) Issue 3
- b) Issue 9
- c) Issue 7

Q5 In which issue is there an article about computer music?
- a) Issue 11
- b) Issue 6
- c) Issue 2

Q6 Optional: In Issue 10 we introduced a new layout style. What is your opinion of the new layout and what could we do to further improve our presentation?

Q7 Optional: what articles or types of article would you like to see more of in The MagPi?

PRIZES

1st The MagPi volume 1 bundle, Adafruit Raspberry Pi Starter Pack - Includes a Raspberry Pi Computer Adafruit Prototyping Pi Plate Kit, Adafruit Pi Cobbler Breakout Kit, Adafruit microSD card adapter Bare Conductive House Kit, Programming Raspberry Pi Book (S. Monk), LEDborg, PicoBorg, BerryClip Pi Supply Kit, Python for Kids book, Quick2Wire full kits (each with interface kit, port expander kit and analogue kit)

2nd Bare Conductive Card Kit, Pi Model B, Case, power supply and SD Card (from RS components), LEDborg PicoBorg, BerryClip, Pi Supply Kit, Quick2Wire full kit (each with interface kit, port expander kit and analogue kit) plO micro SD adapter, Super Scratch Adventure books, Programming Arduino book (S. Monk)

3rd Proto Armour case, Bare Conductive House Kit, LEDborg, PicoBorg, BerryClip Pi Supply Kit, Quick2Wire full kits (each with interface kit, port expander kit and analogue kit) plO micro SD adapter, Python for Kids book, Programming Raspberry Pi book (S. Monk)

4th CPC vesa mount case, Bare Conductive Card Kit, LEDborg, PicoBorg, BerryClip, Pi Supply Kit plO micro SD adapter, Super Scratch Adventure book, Programming Arduino book (S. Monk)

For the complete list visit www.themagpi.com/birthday

SPONSORS

adafruit INDUSTRIES

Bare CONDUCTIVE

Pimoroni

PROTO ARMOUR

cpc

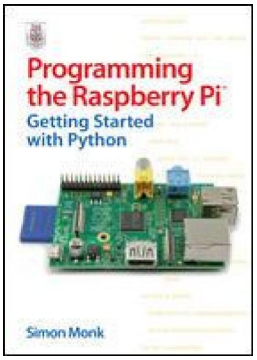
Pi Supply

PiBorg

RS

no starch press

Quick2Wire



Quillix

PCSL

Meltwater

Bash Gaffer Tape

```
if [[ $var == 1 ]]; then
   cat /proc/cpu # Check the CPU type
elif [[ $var == 2 ]]; then
   cat /proc/meminfo # Memory information  else
   date -l # The current date
fi
```

2 - Embedding text and programs

DIFFICULTY : MEDIUM

W. H. Bell

MagPi Writer

Challenge solution

How did you get on with the challenge problem in Issue 10? Here is a solution to the problem,

```
#!/bin/bash
# A script to gzip all of the files in the present working
# directory.  The script does not consider gzip files which
# have the .gz ending.
for file in $(find $PWD -maxdepth 1 -type f | grep -v .gz); do
  gzip $file
done
```

Bash (Bourne Again Shell) provides a lot of functionality, but one of its key strengths is the way other scripting languages or programs can be lashed together to produce a final application.

Embedded files

Open a terminal window and use the nano editor mentioned within the last tutorial session. Then create a new script file containing,

```
#!/bin/bash
cat > input_file.txt <<EOF
ColumnA_row1 ColumnB_row1 ColumnC_row1 ColumnD_row1
ColumnA_row2 ColumnB_row2 ColumnC_row2 ColumnD_row2

ColumnA_row3 ColumnB_row3 ColumnC_row3 ColumnD_row3
ColumnA_row4 ColumnB_row4 ColumnC_row4 ColumnD_row4
EOF
```

save it and make it executable (`chmod u+x script.sh`). Execute the script and look at the `input_file.txt` that it produces. The `input_file.txt` matches the text given between the `<<EOF` and `EOF` statements. For this script to work, there must not be a space after `<<EOF`. The syntax of the script states that the text following `<<EOF` is redirected to the file `input_file.txt` until `EOF` is reached. The redirect operator (>) works in the same way as mentioned in the last tutorial, truncating the output file and then appends to it. The `<<EOF` syntax can be very helpful for embedding parts of configuration files within a script, rather than using many echo commands.

Splitting input lines

If a file is read using a standard `for` loop, then each word (separated by one or more spaces) becomes a value within the loop. This can be helpful for some problems, but the line number information is lost. The `read` command provides a solution,

```
#!/bin/bash
while read line; do
  for word in $line; do
    echo -n "$word "
  done
  echo ""
done < input_file.txt
```

In this example, the outer loop reads each line of the file into the variable `$line`. Then in the inner loop, each word within a line is considered by the `for` loop. The `-n` in the `echo` command means that `echo` does not add a newline character after printing the line. The input file is associated with the `while` loop by using input redirection (<).

Using Perl

Perl is a great language for string manipulation and can be easily used within Bash scripts,

```
#!/bin/bash
while read line; do
  c=$(echo "$line" | perl -ne '@words=split(/\s+/,$_); print STDOUT $words[2];')
  d=$(echo "$line" | perl -ne '@words=split(/\s+/,$_); print STDOUT $words[3];')
  echo $c $d
done < input_file.txt
```

In this case, the outer `while` loop is the same as the previous example. `"$line"` is used to make sure the white spaces are also sent to the perl command. Using the pipe operator causes the output of the `echo` command to be the input for the Perl command. The `-ne` options allow Perl to be embedded in this manner. The `$_` variable contains the value present in the variable `$line`. Then the `split` command splits the words with one or more spaces (`\s+`) into the array `@words`. The second and third array elements correspond to the `c` and `d` columns of the file, where the first column is the zeroth element. Similar to C, each Perl statement is terminated by a semicolon. More information on Perl can be found at http://www.perl.org/.

Once a line has been split into different strings, one of the strings can be used as a signal that the rest of the file should be printed,

```
#!/bin/bash
found=0
while read line; do
  c=$(echo "$line" | perl -ne '@words=split(/\s+/,$_);if($#words > 2) {print STDOUT $words[2]; }')
  if [[ -z $c ]]; then
    continue
  fi
  if [[ $c == "ColumnC_row2" ]]; then
    found=1
  fi
  if [[ $found == 1 ]]; then
    echo $line
  fi
done < input_file.txt
```

This time the Perl part of the script checks the array size and only uses the third element if there are three or more array elements. The first `if` statement ignores zero length `c` variables, which would be caused by there being less than three words in a given line. Then as soon as found is set to one, the rest of the file is printed to the screen.

Using other programs

```
#!/bin/bash
gnuplot -persist <<EOF
set terminal pngcairo
set output 'sin.png'
set key inside right top
set yrange [-1.2:1.2]
set samples 60, 60
plot [-6:6] sin(x)
EOF
```

Several different programs can be embedded within the same script. For example, gnuplot used in previous C Cave can be used to produce a png file illustration of a mathematical function. In this case, the `<<EOF` syntax implies that the commands are passed to the gnuplot program. The `-persist` gnuplot option is used to keep the gnuplot session open, such that it reads the commands after the `<<EOF`. More information on gnuplot can be found at http://www.gnuplot.info/.

The gnuplot script can be used to plot any function given at the command line by replacing `sin(x)` with `$1`. Then add,

```
if [[ -z $1 ]]; then
  echo "Usage: $0 <f(x)>"
  exit 1
fi
if [[ $1 != *"(x)"* ]]; then
  echo "Usage: $0 <f(x)>"
  exit 2
fi
```

before `gnuplot -persist <<EOF`

The first `if` statement protects against a missing argument. The second `if` statement checks that the argument provided is a function of `x`. Try running the script by typing, `./script.sh "cos(x)"`

Challenge problem

Write a Bash script which contains a Python program to calculate mathematical functions. Start from,

```
python <<EOF
import math
from math import *
EOF
```

The solution will be given next time.

antiloquax

evil genius

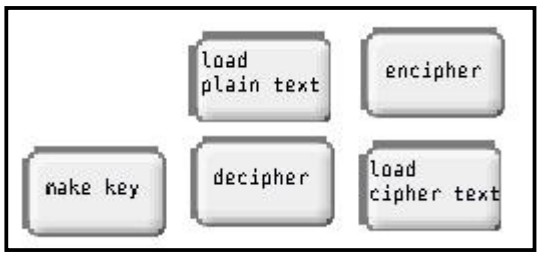
Pretty Poor Privacy

DIFFICULTY : EASY

This program shows you how to implement a "simple substitution" cipher with a reciprocal key. In a reciprocal key, pairs of characters are mapped to each other. This makes life easier as we can use the same key to encipher and decipher messages. The resulting cipher texts are not too difficult to crack, but it's a start!

The first thing to do is make some sprites that will be the buttons we click to make things happen in our program (see the examples on the right).

To create a new sprite, click the "paint new sprite" button just below the stage. You can delete the cat!



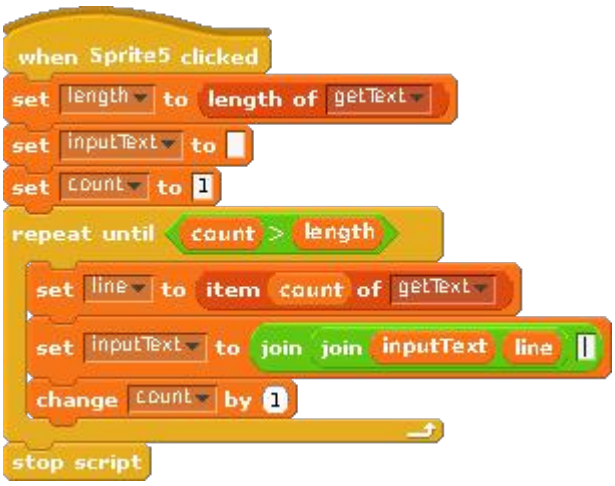
when Sprite4 clicked
delete all of plainAlpha
delete all of cipherAlpha
set alphabet to ABCDEFGHIJKLMNOPQRSTUVWXYZ.,!?
set length to length of alphabet
set count to 1
repeat length
add letter count of alphabet to plainAlpha
change count by 1
repeat length
set index to pick random 1 to length of plainAlpha
add item index of plainAlpha to cipherAlpha
delete index of plainAlpha
stop script

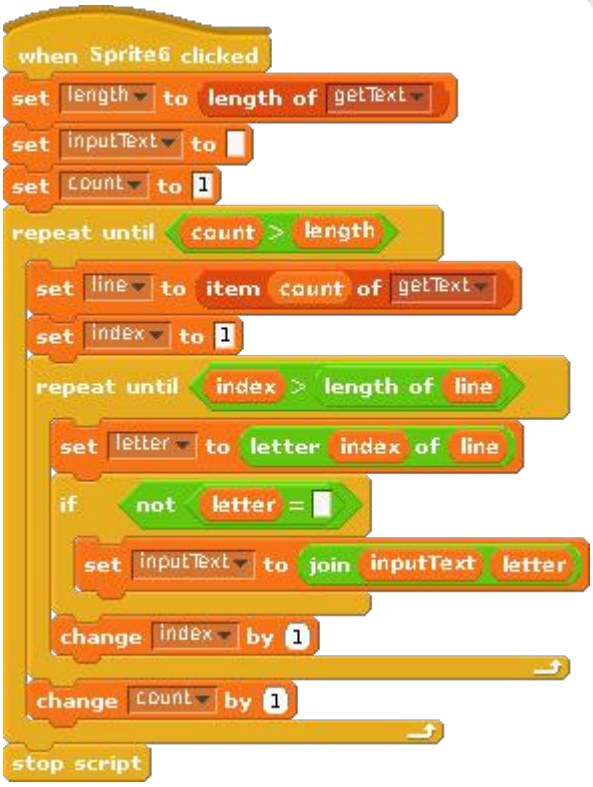
Here's the script that the "make key" button activates (the remaining scripts are on the following pages). To load a key you have already made, make the “cipherAlpha” list visible on the stage, then right-click it and “import” the key from the file in which it has been saved.

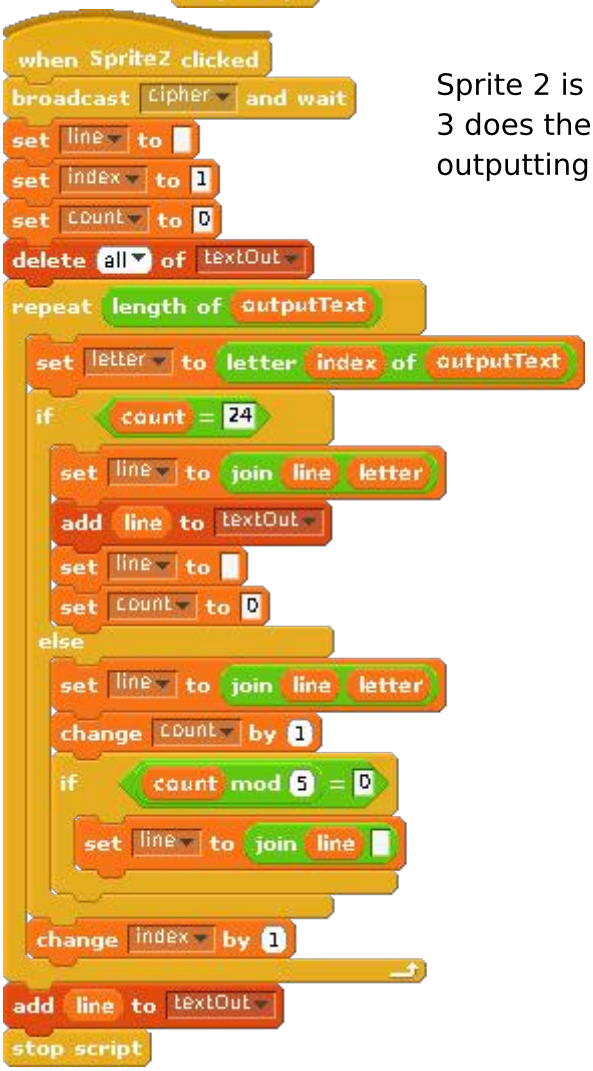
Alternatively, click the “make key” button. It's vital that your contact has a copy of your key so that he/she will be able to decipher your message. Right click "cipherAlpha" and choose "export" to save it to a text file for sharing.

Sprite 5 loads plain text from the getText list into a variable called "inputText". It puts in a "|" character to stand for line-endings.

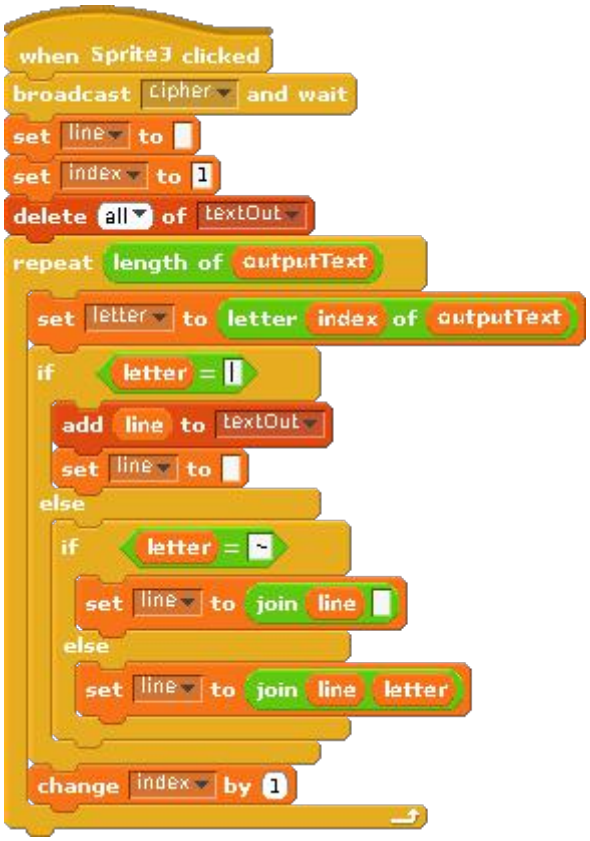
Sprite 6 loads cipher text you've imported into "getText". It ignores any spaces (these are put in to separate the cipher text into 5-character "words").


when Sprite5 clicked
set length to length of getText
set inputText to
set count to 1
repeat until count > length
set line to item count of getText
set inputText to join join inputText line |
change count by 1
stop script


when Sprite6 clicked
set length to length of getText
set inputText to
set count to 1
repeat until count > length
set line to item count of getText
set index to 1
repeat until index > length of line
set letter to letter index of line
if not letter = |
set inputText to join inputText letter
change index by 1
change count by 1
stop script


when Sprite2 clicked
broadcast cipher and wait
set line to
set index to 1
set count to 0
delete all of textOut
repeat length of outputText
set letter to letter index of outputText
if count = 24
set line to join line letter
add line to textOut
set line to
set count to 0
else
set line to join line letter
change count by 1
if count mod 5 = 0
set line to join line
change index by 1
add line to textOut
stop script

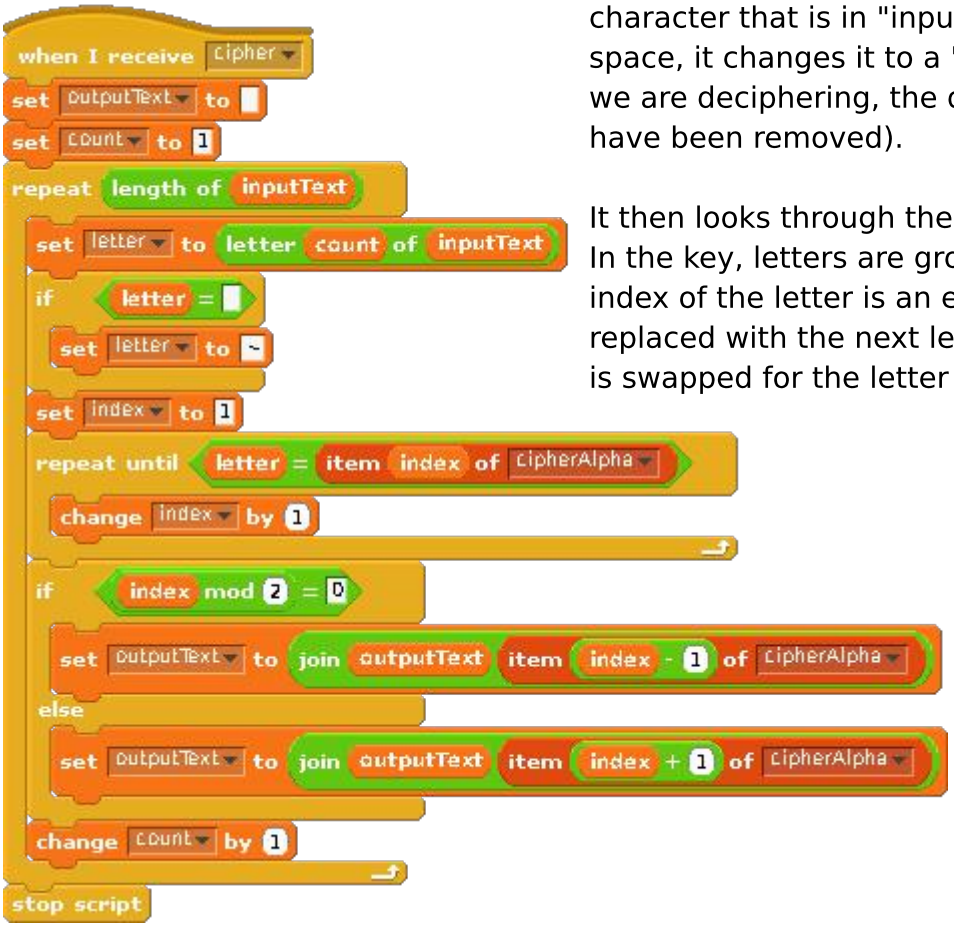
Sprite 2 is run when the "encipher" button is clicked and sprite 3 does the deciphering. Both call the cipher script before outputting the properly formatted text to the "textOut" list.



After calling the "cipher" block, these scripts prepare the text for output. If we are preparing cipher text, it is broken up into lines each of which can have up to five five-character cipher "words".

If we are preparing plain text, any "|" symbols result in a newline being added, while a "~" is changed into a space.

The resulting text is then availble in "textOut" and you can export to a text file if you wish.


when I receive cipher
set outputText to
set count to 1
repeat length of inputText
set letter to letter count of inputText
if letter =
set letter to ~
set index to 1
repeat until letter = item index of cipherAlpha
change index by 1
if index mod 2 = 0
set outputText to join outputText item index - 1 of cipherAlpha
else
set outputText to join outputText item index + 1 of cipherAlpha
change count by 1
stop script

This is the "cipher" script. It just goes through each character that is in "inputText". If the letter is a space, it changes it to a "~" before enciphering it. (If we are deciphering, the dummy spaces will already have been removed).

It then looks through the key until it finds the letter. In the key, letters are grouped in pairs. So if the index of the letter is an even number, the letter is replaced with the next letter in the list. Otherwise it is swapped for the letter before it.

Scratch On!

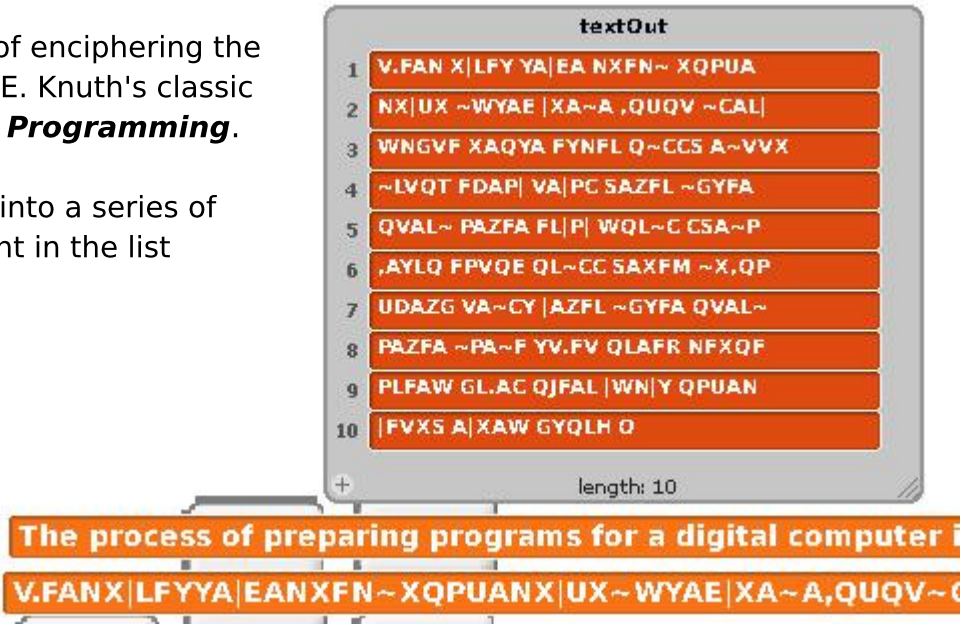
In this example, I've made the "inputText" variable visible so you can see that the message has been loaded.

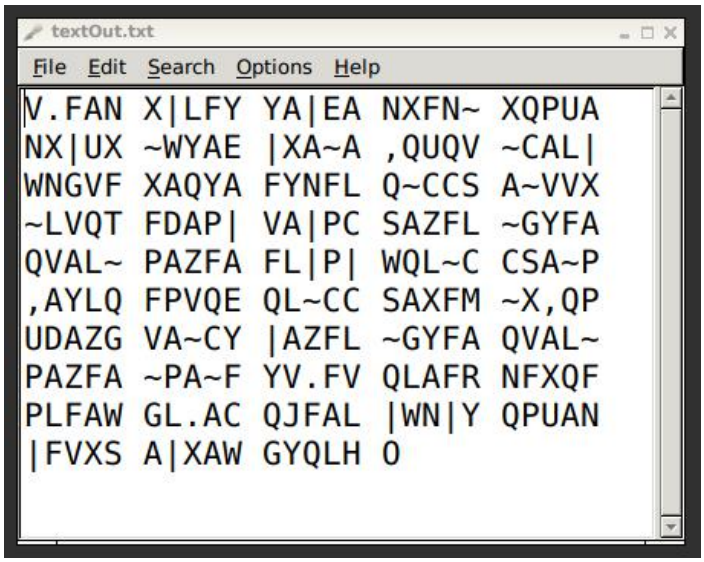
getText
1
The process of preparing programs for a digital computer is especially attractive, not only because it can be economically and scientifically rewarding, but also because it can be an aesthetic experience much like composing poetry or music.
length: 1

The process of preparing programs for a digital computer

Here you can see the result of enciphering the opening sentence of Donald E. Knuth's classic book ***The Art of Computer Programming***.

The cipher text is broken up into a series of lines, each line is one element in the list textOut.


textOut
1 V.FAN X|LFY YA|EA NXFN~ XQPUA
2 NX|UX ~WYAE |XA~A ,QUQV ~CAL|
3 WNGVF XAQYA FYNFL Q~CCS A~VVX
4 ~LVQT FDAP| VA|PC SAZFL ~GYFA
5 QVAL~ PAZFA FL|P| WQL~C CSA~P
6 ,AYLQ FPVQE QL~CC SAXFM ~X,QP
7 UDAZG VA~CY |AZFL ~GYFA QVAL~
8 PAZFA ~PA~F YV.FV QLAFR NFXQF
9 PLFAW GL.AC QJFAL |WN|Y QPUAN
10 |FVXS A|XAW GYQLH O
length: 10
The process of preparing programs for a digital computer
V.FANX|LFYYA|EANXFN~XQPUANX|UX~WYAE|XA~A,QUQV~


textOut.txt
File Edit Search Options Help
V.FAN X|LFY YA|EA NXFN~ XQPUA
NX|UX ~WYAE |XA~A ,QUQV ~CAL|
WNGVF XAQYA FYNFL Q~CCS A~VVX
~LVQT FDAP| VA|PC SAZFL ~GYFA
QVAL~ PAZFA FL|P| WQL~C CSA~P
,AYLQ FPVQE QL~CC SAXFM ~X,QP
UDAZG VA~CY |AZFL ~GYFA QVAL~
PAZFA ~PA~F YV.FV QLAFR NFXQF
PLFAW GL.AC QJFAL |WN|Y QPUAN
|FVXS A|XAW GYQLH O

Here you can see what the message looks like after it has been exported and opened in leafpad.

If you get stuck, the program is available here:
http://tinyurl.com/scratch-subs

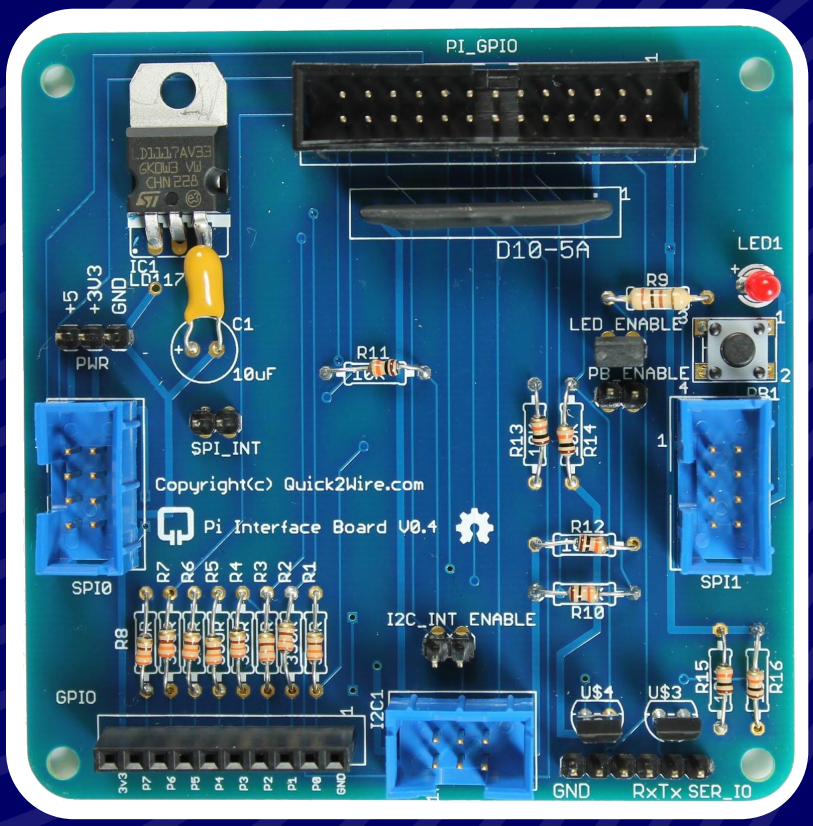
Here's the key I used in my example, in case you want to try deciphering the whole thing. Remember, it is a reciprocal key. Letters at odd numbered indices map to the next letter, while letters at even numbered indices map to the letter before. Scratch lists begin with index 1. So, 'K' is at index 3 and maps to 'J' at index 4. And 'J' maps back to 'K'.

```
O|KJTVFEIQ~ARX,DYSPN.HZBMWGUCL
```

If you are interested in learning more about cryptography, try this website:
www.simonsingh.net/The_Black_Chamber/

Quick2Wire

SAFE AND SIMPLE CONNECTION TO YOUR RASPBERRY PI



HARDWARE

- Interface board
- I²C Port Extender
- Analogue board
- and more...

SOFTWARE

- For GPIO, I²C, SPI
- Device Libraries
- Examples

SUPPORT

- Articles
- Tutorials
- Forum

New kits and more in our store

Find out more at quick2wire.com

Feedback & Question Time

There are people in some amateur radio clubs who can build kits for little or no cost for handicapped people. I personally have built kits in the past for people with disabilities and in a "shoe on the other foot" story a friendly ham is building a kit for me due to a fine motor problem and vision issue.

Steve Black

[ed: In response to Paula Thomas' comment in Issue 11.]

I have downloaded (several times) The MagPi Issue 11 PDF and I find that the contents page, I presume, and first article on controlling your heating has no text within the page. Is this an error or a problem with me reading the PDF file?

Gary

[ed: We've had a number of comments from people who have had a few problems viewing Issue 11 in alternative PDF viewers. We would recommend the magazine is read using Adobe Reader to get the best results. We are working to fully resolve the problem, and have fixed several similar issues in Issue 11.]

God love ya. Being that you have these available I guess a donation is in the works as I can't see how these could be provided w/o some sort of access fee. This is what is great about this community and my hat is off to you guys in what you are providing for the community.

Bill Raub

I have had my new Pi for about two weeks now and it is great.

David

I'm a regular reader of the MagPi, and I like the varied articles and projects featured in each issue.

James Medd

I think that you should have a listing of articles that have been written in The MagPi on your website. Now, if a user wants to check out what has been written or get back to some article he has read before, there is (or I could not find) no way to look at those, other than opening each issue separately.

I think this would enhance user experience quite a lot. One way to do this would be to have issues' tables of content typed out on a page (call that "Issues" or whatever).

Another feature could be that these TOCs could be interactive in the sense that one could open a pdf that consists of just the pages with that article (or letters to the editor or other content), or the Issuu viewer opened at the correct page.

Mikko Kouhia

[ed: We do have something similar in mind for the future.]

The MagPi is a trademark of The MagPi Ltd. Raspberry Pi is a trademark of the Raspberry Pi Foundation. The MagPi magazine is collaboratively produced by an independent group of Raspberry Pi owners, and is not affiliated in any way with the Raspberry Pi Foundation. It is prohibited to commercially produce this magazine without authorization from The MagPi Ltd. Printing for non commercial purposes is agreeable under the Creative Commons license below. The MagPi does not accept ownership or responsibility for the content or opinions expressed in any of the articles included in this issue. All articles are checked and tested before the release deadline is met but some faults may remain. The reader is responsible for all consequences, both to software and hardware, following the implementation of any of the advice or code printed. The MagPi does not claim to own any copyright licenses and all content of the articles are submitted with the responsibility lying with that of the article writer. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit:

http://creativecommons.org/licenses/by-nc-sa/3.0/

Alternatively, send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

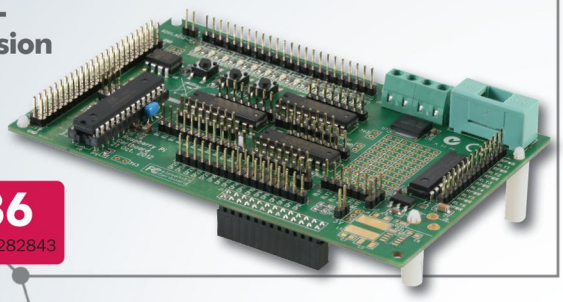
Project Builder?

Connect your Raspberry Pi to the World

Visit cpc.co.uk for full product details

Gertboard

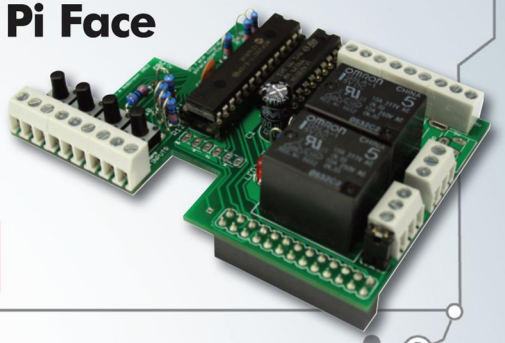
GPIO - Expansion Board



£36
code SC1282843

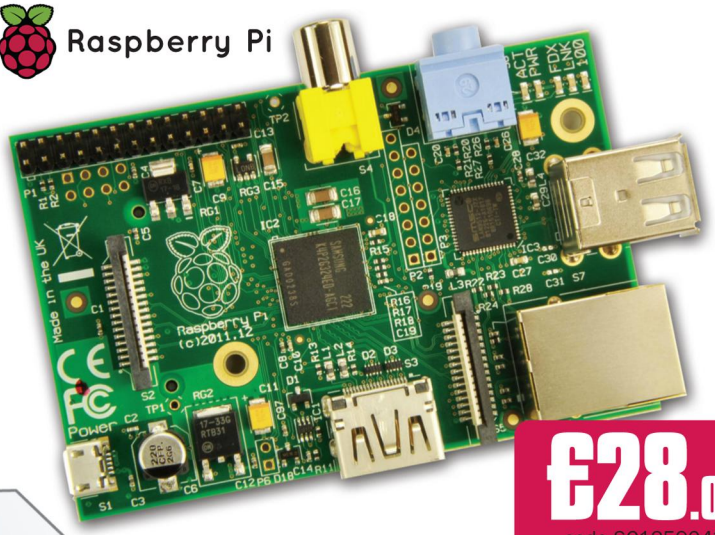
Digital Pi Face

GPIO - Expansion Board



£24
code SC1282743

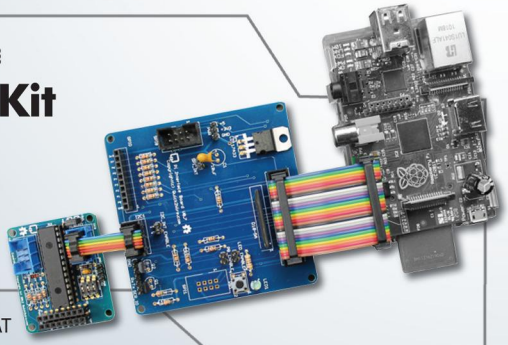
Raspberry Pi



Made in the UK

£28.07
code SC1259043

Quick2Wire Combo Kit

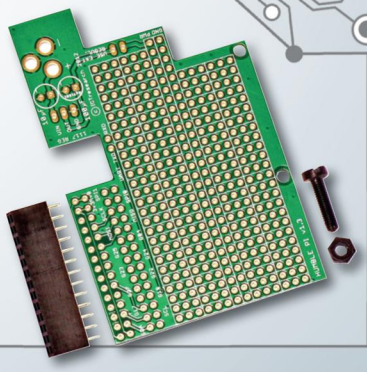


£22.74
code SC1294143

All prices include VAT

CISECO Humble Pi

Prototyping Board



£4.91
code SC1287143

cpc.co.uk

For Everything Raspberry Pi...

- Kits and Cases
- Expansion, Breakout and Prototyping Boards
- Vast Range of Components
- WiFi Dongles & HDMI to VGA Adaptors
- Video and Audio Leads
- Keyboards, Mice and Monitors
- GPIO Ribbon Cables
- Programming and User Guides

FREE DELIVERY

Conditions apply, visit cpc.co.uk